1973-06

# A helicopter flight performance system using an LSI microprocessor

Eloe, Edwin Eugene; Scott, Richard Tazewell, Jr.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/16640

# A HELICOPTER FLIGHT PERFORMANCE SYSTEM USING AN LSI MICROPROCESSOR

Edwin Eugene Eloe

# NAVAL POSTGRADUATE SCHOOL

Monterey, California

# THESIS

A HELICOPTER FLIGHT PERFORMANCE
SYSTEM USING AN LSI MICROPROCESSOR

by

Edwin Eugene Eloe
and
Richard Tazewell Scott, Jr.

Thesis Advisor: Gary A. Kildall

June 1973

# A Helicopter Flight Performance System Using an LSI Microprocessor

by

Edwin Eugene Eloe
Captain, United States Marine Corps
B.S., University of Utah, 1966

and

Richard Tazewell Scott, Jr.
Lieutenant, United States Navy
B.S., United States Naval Academy, 1967

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1973

# ABSTRACT

This thesis presents the development of a helicopter gross weight calculator. Required qualities for aircraft system components such as durability, reliability and low weight are met using an LSI micro-processor. The prototype system which was developed weighs approximately four pounds and has approximate dimensions of 7" x 5" x ½". The cost estimate for the system is less than $350. The calculator solution is based on the solution technique currently being used by Naval Aviators which is obtained from nomographs in the aircraft NATOPS Manual. Minor modifications are required to make this system applicable to different helicopter types. A listing of the calculator program and a discussion of the prototype's operation are included.

TABLE OF CONTENTS

## LIST OF FIGURES

## ACKNOWLEDGEMENTS

We wish to express our sincere appreciation to Ray Brubaker, who spent so much time and effort building the hardware interface equipment between the keyboard and the display LED's.

The authors are deeply grateful to Ken Farrell of USAATA for his assistance in explaining the alternate solution being tested by the Army and Air Force.

Last, but not least, we would like to thank Gary Kildall, our thesis adviser, for his advice and guidance during the course of our work.

# I. INTRODUCTION

Each type of operational helicopter employed by the Navy and Marine Corps has an associated Naval Air Training and Operating Procedures Standardization (NATOPS) Manual. One section of the NATOPS Manual is comprised of a set of charts which may be used by aviators to determine specific values to measure the aircraft flight performance capabilities. In particular, one chart provides the aviator with a value for the maximum allowable gross weight for which the helicopter may be transitioned from forward flight to a hover without the contribution to lift capability provided by an associated "ground cusion" (Hover Out of Ground Effect). The HOGE Charts of the NATOPS Manual are in fact the primary method presently available to Naval Aviators for determining a predicted gross weight to hover. This parameter is considered critical to helicopter commanders for all operational flights. A standard aircraft mission for any given day may dictate variations in aircraft configuration due to "payload" changes and fuel consumption. These variations are generally not condusive to prescheduling. Atmospheric conditions and altitude requirements are subject to significant changes during any twenty-four hour period. These factors all contribute to necessitate continual updates of lifting capability.

The HOGE charts are generally nomographs which represent a compilation of test data for a given aircraft type. Input parameters to the nomograph are pressure altitude, ambient

air temperature and wind speed. To determine the gross weight hovering capability, the pilot enters the associated nomograph with a value for pressure altitude. The intersection of a line horizontal from the pressure altitude with the ambient temperature line projected vertically downward to the gross weight scale yields a value for the no-wind gross weight. Determination of the temperature line, unless it coincides with a range boundary, requires an interpolation process. The standard procedure for this process is to assume that all temperature lines are parallel to their closest boundary temperature line and that the distance between each successive temperature line is equal. The no-wind gross weight may be adjusted to account for wind speed. The gross weight determined above is used as an input to the second part of the nomograph. The graph is entered vertically with the no-wind gross weight value. This point is moved parallel to its closest wind line until it reaches a line corresponding to the input wind speed magnitude. The vertical projection of this intersection to the weight scale is the maximum gross weight with which a helicopter can hover under the given atmospheric conditions.

A sample solution is shown in Figure 1. At point 1, the pilot enters the nomograph with a pressure altitude of 10,000 feet. He then moves horizontally to point 2, the ambient temperature line of 15 degrees. Projecting point 2 vertically to point 2b produces the no-wind gross weight of 7000 pounds. To correct for a wind speed of twenty knots, the graph is

entered at point 2a with the no-wind gross weight value. This point is moved parallel to its closest wind line as shown to point 3 where the wind line intersects the head-wind line associated with twenty knots. The vertical projection of point 3 to point 3a produces a gross weight value of 7875 pounds.

Before any landing or take off is made in a Naval helicopter, the aircraft commander is expected to insure, using a nomograph in the NATOPS Manual, that the maneuver is executable with respect to aircraft weight limitations. Assuming he knows his present gross weight, he subtracts it from the computed gross weight to determine if he can land. If the number is negative, he should not land. If it is zero, he can land but cannot increase the gross weight before taking off again. If it is a positive number, that is how much of a load he can safely take on.

Many times what actually happens is that either the pilot does not have a NATOPS Manual in the aircraft or he doesn't use it. It is much easier to rely on experience and guessing than to read the graph. The size of the NATOPS Manual tends to minimize its use in an aircraft cockpit. Also, some interpolation of the HOGE charts is required. Without a certain degree of care in the interpolation process, the resulting critical parameter will contain significant error. Aircraft vibration plus cockpit responsibilities are factors which may restrict effective chart interpolation in a cockpit environment. The pilot,

SAMPLE SOLUTION

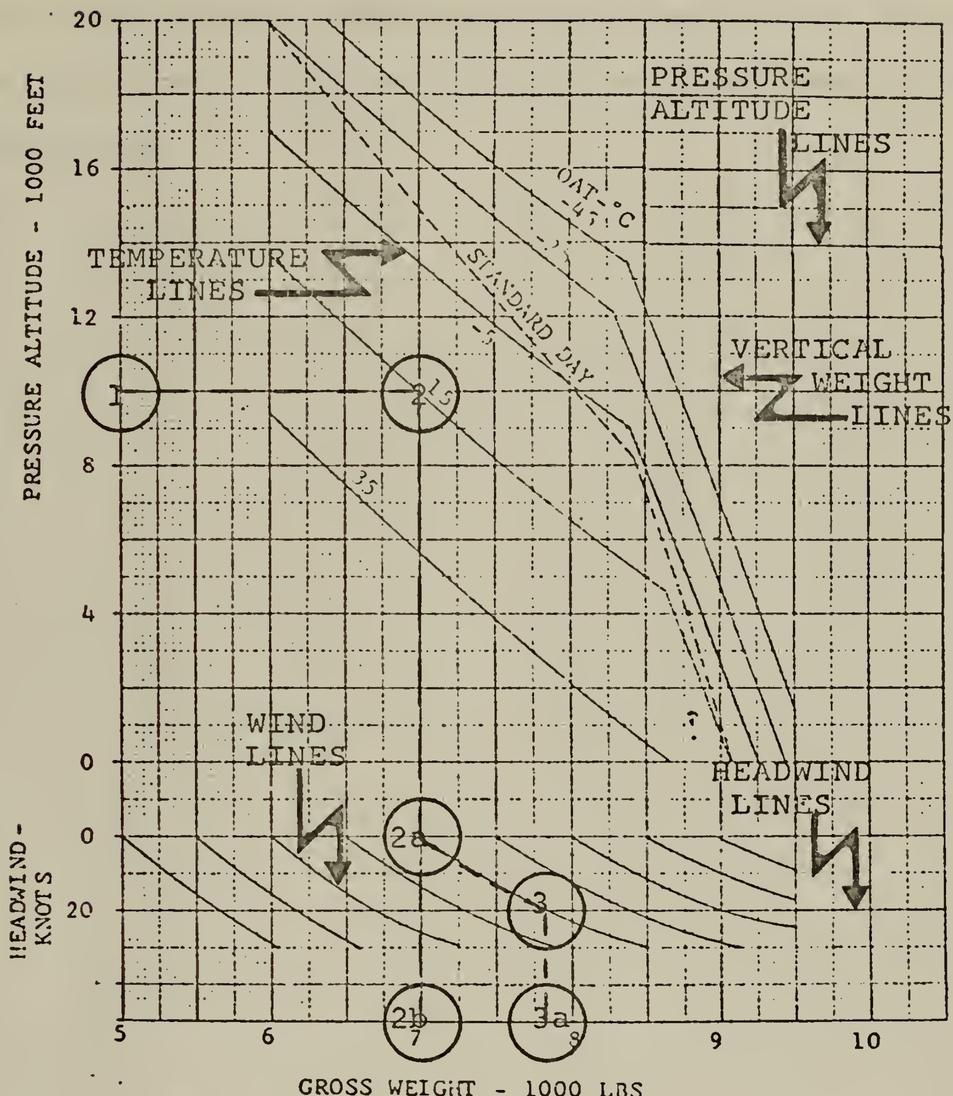



FIGURE 1

therefore, may attempt to land or take off when the graph should have told him it would be impossible. The Navy Safety Center states that 10% of helicopter crashes are caused by overweight conditions and they suspect that the figure could be as high as 40% if the actual causes could be determined.

## II. ALTERNATE SOLUTIONS

In light of the previous discussion, it is evident that an improved technique for determining the maximum gross weight which will still permit a hover must be developed for cockpit use.

Under the Joint Army-Navy Aircraft Instrument Research (JANAIR) sponsorship, Transonics Incorporated has developed a Helicopter Lift Margin System (HLMS) described in reference 1, which is intended to fill this cockpit requirement. Flight tests are currently being conducted by the United States Army Aviation Systems Test Activity, USAASTA, at Edwards Air Force Base, to evaluate the effectiveness of Transonic's HLMS. It is claimed that this system will compute the gross weight of the aircraft using analog input from the engine and rotors. The engine inputs will tell how efficiently the aircraft engine is performing. Rotor input will tell how efficiently the rotor system is lifting the aircraft depending on the outside air temperature and air density at that time. The pilot would input headwind and altitude. Once the gross weight is computed, the theoretical maximum gross weight computed for the given conditions is computed. The actual gross weight is then subtracted from the theoretical maximum gross weight. The result is the lift margin for that specific aircraft on that specific time of day with those specific atmospheric conditions.

Another system developed by Marconi Elliott Avionics Systems Ltd. of Great Britain provides the pilot with a power margin indication in terms of a ratio between the present engine power output to maximum power output. The Elliott Helicopter Energy and Rotor Management System, HERMES, of reference 2, expects pilot input of temperature, static air pressure and helicopter gross weight. The temperature and pressure are used to compute engine power output based on data from Briston-Siddeley Technical Specification Number TSC 160 applicable to the GNOME H 1400. Using computed power output, the maximum weight lifting capability is computed. This value is then compared with the weight input to determine the power margin. The HERMES system is compact and weighs approximately eighteen pounds. The accuracy and reliability of this system as well as the general use potential and cost are not yet available.

The advent of micro-computers with their many favorable qualities such as low cost, high durability, low power requirements, miniature size and light weight, along with the fact that they are programmable, has produced a need to examine the potential of these computers in systems requiring these properties.

The fact that the previously mentioned systems have been developed is sufficient to justify the obviousness of the need. Caution should be exercised with the hardware selection to insure that the operational computer will best meet the system requirements.

The intent of this thesis is to investigate the suitability of the use of a micro-computer as an alternative method of solution to system requirements of the type described. For this investigation, the primary considerations are the following: the system must possess an acceptable degree of accuracy, it must produce usable output, it must be adaptable to a helicopter cockpit, and it must be competitively priced. The application discussed here uses an MCS-4 micro-computer built by Intel, described in Reference 3, as the basis for the design of a prototype system which will accept a set of input parameters and produce the gross weight capabilities of a helicopter as output.

## III. SOLUTION

The Nomographs which are presently used by Naval Aviators to provide aircraft performance limits were divided into regions as indicated in Figure 3. The curve segments within each region are represented in the solution by linear approximations of those segments using the end points of each segment to determine the respective slope and intercept. The slope of a given segment is not necessarily equal to the slope of the preceding segment in that region; therefore, slopes are adjusted using the differential slope between two segments in a region and the associated differential temperature. For this application, the temperature differential between any two segments is assumed to be linear. The computed values of slope and intercept for each segment within each règion are stored in a data table.

The input temperature dictates the starting position in the data table. A higher likelihood exists that the computed value of gross weight will fall in region II or region III based on standard day temperature and normal altitude requirements. For this application, the first computation uses coefficients associated with segments in region II rather than region III. This eliminates the possibility of three iterations to compute the no-wind gross weight. Once the initial value for gross weight is determined, the magnitude is compared to a test value which separates region II from region III. In the event that the value for gross

weight is less than the test value, a second test is made to determine if the gross weight is in region I. If either test condition is met, the coefficients for that region are used to compute a new no-wind gross weight. If neither test condition is met, the computed value is the no-wind gross weight.

The value of no-wind gross weight is used to determine the entry point into the wind speed correction data table. The seven wind speed regions are each segmented at intervals representing ten knots of wind speed. Coefficients corresponding to segment slopes are loaded from the data table to adjust the gross weight for wind. The wind speed adjustment is made in increments of ten knots. The coefficient is updated after each computation and input wind speed is reduced by ten knots. The final adjustment for wind speed is made when the current value of wind speed is between zero and ten knots. The resultant is output as the maximum gross weight for HOGE relative to the set of input parameters. A device which will accept the indicated input parameters, operate on those parameters, and output a value for the maximum gross weight was constructed. The device hereafter referred to as the Helicopter Gross Weight Indicator, HGWI, employs components available to students at the Naval Postgraduate School micro-computer laboratory. The hardware for the HGWI prototype consists of an MCS-4 computer interfaced with an input-output device. The software is comprised of a data structure, a set of subroutines and a main program.

The data structure is the section of software which is aircraft type dependent. It consists of a table or set of tables in read-only-memory (ROM) containing constants extracted from a specific nomograph. The HH-IK helicopter was arbitrarily selected as the representative aircraft for development of the data structure in this particular HGWI. Further references to this particular system use the designation HGWI/HH-IK.

The main program is an algorithm which, using a set of subroutines and the data structure, operates on the input parameters to compute a solution for output by the HGWI. The hardware, data structure, subroutines and main program are further described in subsequent sections.

## A. HARDWARE

The MCS-4 micro-computer set [Ref. 3] consists of the following three chips, each packaged in a sixteen pin DIP package:

(1) A Central Processor Unit Chip (CPU 4004)

(2) A Read Only Memory Chip (ROM 4001)

(3) A Random Access Memory Chip (RAM 4002)

The prototype also includes a keyboard type input-output device with its associated electronics. This keyboard in the production model would have to be modified to eliminate the illegal keys and excess Light Emitting Diodes.

### 1. 4004 CPU

The CPU contains the control unit and the arithmetic unit. It works in conjunction with the 4001 ROM'S and the

4002 RAM'S and the I/O package to form a completely self contained system. The CPU communicates with the RAM'S and ROM'S of the system through a four line data bus and with the I/O package through ROM ports.

2. 4001 ROM

The 4001 ROM is a 2048 Bit Metal Mask programmable ROM. Each chip is organized as 256 x 8 bit words which contain the program and data table. Each chip also has a four bit input-output (I/O) port which is used to route information to and from the data bus lines in and out of the system. The prototype uses eight 1702 programmable ROMS. The 1702 is equivalent to the 4001 ROM except that the 1702 may be erased and reused. Seven 4001 ROMS would be used in the production model.

3. 4002 RAM

The 4002 RAM stores three hundred twenty bits arranged as four registers of twenty-four bit characters which are divided into sixteen main memory characters plus four status characters. The prototype contains four 4002 RAM chips.

Input is accomplished by a standard mechanical type keyboard in the prototype. Output is through sixteen light emitting diode digits which are connected to the CPU through a shift register and a decoder/driver, as in Figure 2.

When a key on the keyboard is depressed, this mechanically sets up a BCD code at the output of the keyboard. This code contains one extra bit which tells whether the

depressed key is a number or a control key. If the key is a number, it is transferred, by software, to a designated position in RAM register fifteen. If it is a control key, program control is transferred to the section of the program which will correctly process the key. Since all keys on this particular keyboard are not required, all illegal keys are ignored. Repeatedly during the input process, a routine is called to display RAM register fifteen. This display routine sets one bit of the shift register. This bit is shifted at every clock pulse. As the bit is shifted, the corresponding LED is enabled. The number which is displayed on the enabled LED is the input to the decoder/driver. This input comes from the enabled position of RAM register fifteen. The enabled position of RAM register fifteen is controlled in the same manner as the enabled LED. Each position of RAM register fifteen will contain either a BCD zero through nine or a fifteen. BCD digits are displayed on the enabled LED as decimal digits, while a fifteen is displayed as a blank.

Because the shifting is so rapid, it appears to the user that the whole decimal number is being displayed at one time. In actuality, only one digit is being displayed at a given instant by the DISPLAY routine.

## B. DATA STRUCTURE

Two tables of data are required for the HGWI/HH-IK. One table is a four by fifteen array which contains the constants required to compute the no wind gross weight.

There are four twenty-degree temperature ranges depicted in Figure 3, where temperature units are degrees centigrade. Range 1 is from $-45^{\circ}$ to $-25^{\circ}$, range 2 is from $-25^{\circ}$ to $-5^{\circ}$, range 3 is from $-5^{\circ}$ to $15^{\circ}$ and range 4 is from $15^{\circ}$ to $35^{\circ}$. The columns of Table 1 correspond to these temperature ranges. Each temperature range is divided into three regions to improve the linearity assumption. Region I is separated from region II by the 7,500 pound gross weight line. Region II is separated from region III by a line connecting the points where the change in slope of the two lines describing a range is maximum. Therefore, each column of Table I contains the coefficients needed to describe four line segments; one within each region and one which separates region II and region III. The second table is a seven by three array. The gross weight for the HGWI/HH-IK may range from 6,000 to 9,500 pounds. The curves which describe the correction for wind speed extend from each 500 pound increment along the gross weight line. The seven increments correspond to the seven columns of Table II. Again, to improve accuracy, the curves of each increment are defined at ten knot intervals. The HGWI/HH-IK corrects for wind speeds up to thirty knots; therefore, there are three rows per column of Table II.

The constants stored in Table I represent the slopes, differential slopes, intercepts and differential intercepts of the line segments as described above. Table II consists of the slopes of the line segments used to correct for wind speed. All constants are stored in a single ROM

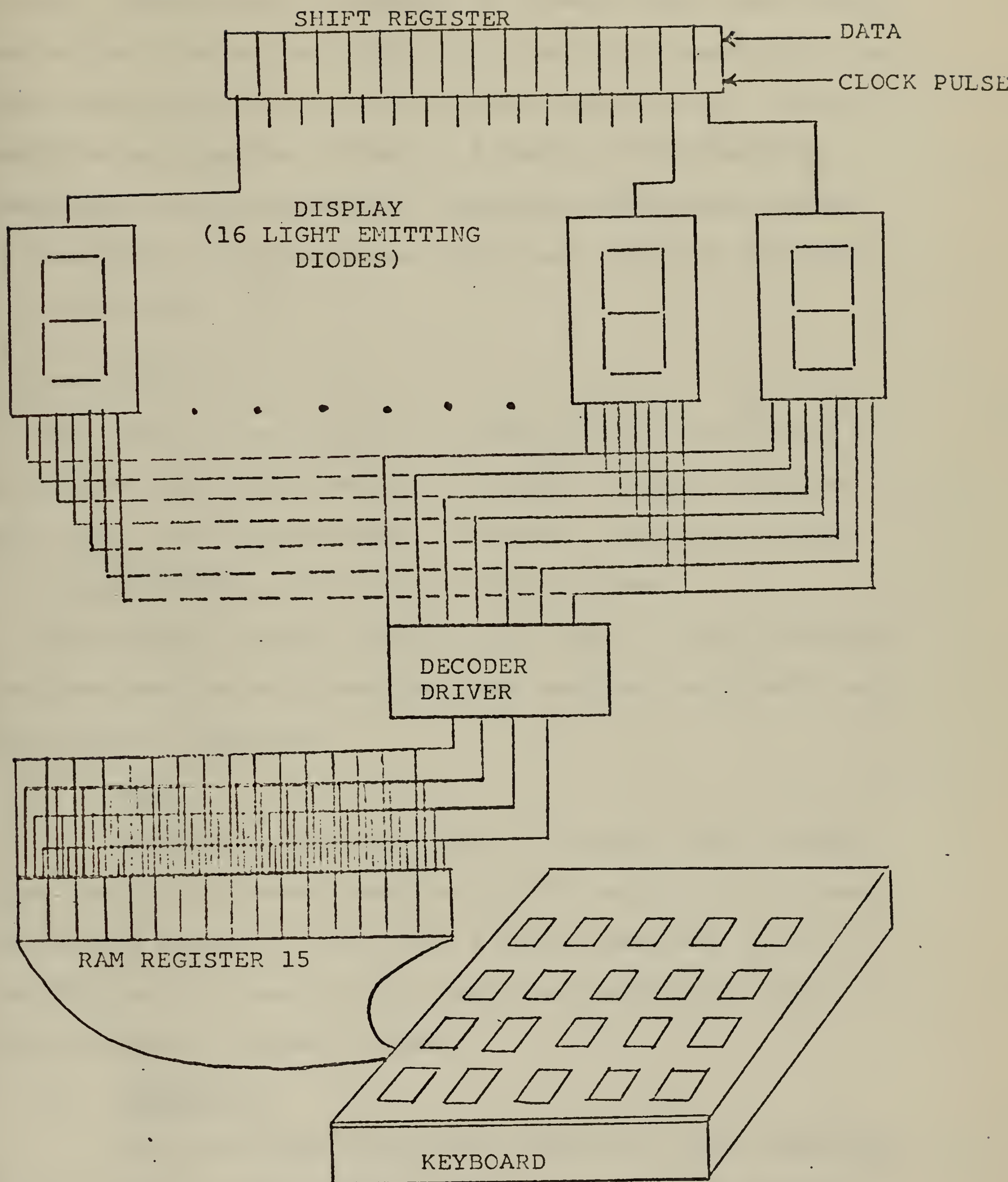

SHIFT REGISTER
DATA
CLOCK PULSE
DISPLAY
(16 LIGHT EMITTING
DIODES)
DECODER
DRIVER
RAM REGISTER 15
KEYBOARD

FIGURE 2.

chip. Two eight bit locations are required per constant stored. Each number is stored in reverse order since this technique is more compatible with the MCS-4 instruction set. The upper four bits of each constant are used to indicate positive or negative numbers. A zero in this location indicates positive numbers. Negative numbers are stored in the ROM in tens complement form and the sign character is set to one.

## C. SUBROUTINES

The subroutines used in the HGWI/HH-IK Calculator can be divided into three main categories: general purpose, arithmetic, and input-output routines. Appendix C gives a complete listing of the calculator program.

The general purpose routines are used by the arithmetic routines and the main program, and perform tasks which are consistently repeated.

### 1. MOVE

The MOVE subroutine takes a sixteen digit number from the RAM register specified by CPU register pair ten-eleven and moves it along with status character zero to the RAM register and status character specified by CPU register pair fourteen-fifteen.

### 2. COMPLEMENT

The subroutine COMPLEMENT takes the Tens complement of the sixteen digit number which is in the RAM register specified by CPU register pair fourteen-fifteen.

3. <u>SHIFLT</u>

The SHIFLT routine takes the sixteen digit number which is in the RAM register specified by CPU register pair twelve-thirteen and shifts it left the number of decimal places specified by CPU register fifteen. The number in CPU register fourteen must agree with the number in CPU register twelve.

4. <u>ZERORR</u>

ZERORR takes the RAM register specified by CPU register pair fourteen-fifteen and places zeroes in each of its sixteen positions and sets its status characters to zero.

5. <u>SHIFRT</u>

SHIFRT is used to take a specified RAM register and shift it right a specified number of decimal digits. The programmer is responsible for making sure that the correct digits of the correct register are specified. SHIFRT assumes that CPU register pair twelve-thirteen contains the address of the rightmost digit of the desired RAM register. In addition, SHIFRT insures that CPU register pair fourteen-fifteen contains the address of the rightmost digit in the desired RAM register minus the number of places to be shifted.

6. <u>SWITCH</u>

SWITCH is a data manipulation routine which takes two eight bit words from the ROM address specified by CPU register pair zero-one. The first one-and-a-half words are the desired number in BCD. These three BCD digits are stored

in the first three positions of the specified RAM register. The last half of the second word in RAM is written into status character zero. If this is an odd number, it indicates the number is negative and this number is in tens complement form. As a result, nines must be placed in positions three to fifteen of the designated RAM register.

7. INIT

INIT initializes the CPU by placing zeroes in CPU registers two through fifteen, a one in register twelve, and a two in register fourteen. INIT also clears both the accumulator and the carry.

8. LOAD

Subroutine LOAD is required because of an idiosyncrasy of the machine. With the MCS-4 one cannot fetch indirect (FIN) from ROM unless the FIN instruction is on the same chip as the constant to be fetched. LOAD transfers an eight bit word from ROM to CPU registers two and three.

9. INITC

INITC is a routine which initializes the CPU registers with constants which are needed to compute the address of entry into a table of constants. These constants are particular to an aircraft type; therefore, they are stored at the base of data Table I.

10. ASCOL

After INITC has loaded into CPU register four the RAM location which contains the computed no-wind gross weight, ASCOL is called to inspect the contents of that RAM. The

column of Table II which contains the constants of interest is numbered corresponding to the number of 500 pound increments above a given base value. ASCOL sets a flag by incrementing CPU register eleven if the weight value in the designated RAM location has a hundreds digit of five or more. The value of the thousands digit is then loaded into CPU register nine. Control is then returned to the calling program.

11. COLNUM

When COLNUM is called, INITC will have loaded a base value in CPU register six, and either TEMPCOL or ASCOL will have loaded CPU register nine with a value to be compared to this base. The base value is the minimum value of temperature if access of Table I is desired or the minimum thousands digit on the gross weight scale if accessing Table II. COLNUM compares register nine with register six and increments the register six value until it is greater than the value in register nine. By fixing the value of the increment and counting the number of increments made, the column number of Table I or the number of thousand pound increments which will be used to determine the column number of Table II, is computed and stored in CPU register eight.

12. COLNUMAS

The subroutine COLNUMAS computes the column number of Table II. COLNUMAS doubles the value in CPU register eight which was returned from COLNUM and adds CPU register eleven which was set by ASCOL. This result is left in CPU register eight.

13. <u>LOADROBAS</u>

This routine loads from the table of constants the number of rows per column in a given table and stores the value in CPU register seven. The base address of the desired table of constants is loaded into CPU register pair zero-one.

14. <u>ASCHECK</u>

To insure that the input value of wind speed is acceptable, ASCHECK inspects the wind speed RAM location. Values of wind speed greater than thirty knots cause ASCHECK to initiate the error routine.

15. <u>TEMPCOL</u>

Prior to calling TEMPCOL, the RAM location of the scaled input temperature will be loaded into CPU register pair four and five by INITC. TEMPCOL checks the value in that location to insure the input temperature is greater than $-45^{\circ}$. If not, the value is outside the range of this application and an error condition results. If the input temperature value is acceptable, TEMPCOL loads the tens digit of that value into CPU register nine and increments it by one if the value of the units digit is greater than zero. The number returned in CPU register nine will be referenced by COLNUM to determine the column number of Table I to access.

16. <u>ADDRS</u>

This subroutine is called after the CPU registers have been initialized. The column number of interest will

have been loaded into register eight by TEMPCOL and COLNUM for Table I or by ASCOL, COLNUM and COLNUMAS for Table II. The table base address was loaded into register pair zero-one and the value of the number of rows per column was loaded into regiser seven by LOADROBAS. ADDRS then computes the address of entry into the respective table of constants by adding to the table base address the product of the column number and the number of rows per column. Control is returned to the calling program with the required address in register pair zero-one.

17. TESTPA

The input value of pressure altitude is inspected by TESTPA. Values greater than 20,000 result in an error condition.

The arithmetic routines are used in the computation of the problem. All negative numbers are in tens complement form with an odd number stored in status character zero. This status character is an easy means of detecting negative numbers. The procedure is to read the status character into the accumulator, clear the carry bit and then rotate the accumulator right one bit. If the carry is on, the number is negative; if it is not on, the number is positive.

18. ADDIT

An arithmetic routine which takes two sixteen digit numbers in RAM registers zero and one and adds them together. The result is in RAM register zero.

19. SUBTRACT

Subroutine SUBTRACT takes a sixteen digit number in RAM register one and subtracts a sixteen digit number in RAM register zero from it. If the number in RAM register zero is negative, the routine recomplements it and adds it to the number in RAM register one. The resulting number is always in RAM register zero.

20. MULT

MULT is a six digit multiply routine using the shift-and-add algorithm. MULT calls SETUP which determines whether the answer will be a positive or a negative number. This is determined by adding the status characters of the two numbers. If the result is an odd number, the multiplication will yield a negative number. If it is an even number, the resulting multiplication is positive. The result of the multiplication is in RAM register zero.

Input-output control of the program is accomplished through CHECKNOSTROBE. The program remains in a loop as long as there is no key on the keyboard depressed. Inside this loop is a call to subroutine DISPLAY which displays the sixteen digit contents of RAM register fifteen. When a key is not depressed, ROM port 1 is read as a one and the loop continues. As soon as a key is depressed, control of the program is transferred to CHECKSTROBE. CHECKSTROBE uses the five-bit key code to index a table called KEYTAB. KEYTAB consists of thirty two jump unconditional instructions each of which corresponds to a legal key or an illegal key.

There are twenty seven keys on the prototype, sixteen legal keys and eleven illegal keys. Control is transferred to the correct code for the depressed key through these jump unconditional instructions in KEYTAB. There are seven routines for legal keys. An illegal key merely transfers control back to CHECKNOSTROBE. The legal keys are described below.

1. <u>Number</u>

The code associated with a number key takes a decimal digit and stores it into RAM register fifteen in the next available digit position. Control is then transferred back to CHECKNOSTROBE.

2. <u>Temperature</u>

The code for the temperature key takes the decimal number in RAM register fifteen, adds forty-five to it as a scale factor and places the result in RAM register A. Control is transferred to OUTCLEAR which loads fifteen into RAM register fifteen. This character displays as a blank. OUTCLEAR then transfers control to CHECKNOSTROBE.

3. <u>Speed</u>

When the speed key is depressed, this code transfers the decimal digit of RAM register fifteen to RAM register six, then transfers control to OUTCLEAR which, in turn, transfers control to CHECKNOSTROBE.

4. <u>Altitude</u>

This key transfers the decimal digits in RAM register fifteen to RAM register five and, after shifting left two places, control is transferred back to CHECKNOSTROBE through OUTCLEAR.

5. <u>Negative</u>

This code allows negative temperatures to be placed into the prototype. If the number is negative when the temperature key is depressed, the contents of RAM register fifteen is complemented and then the forty-five scale factor is added.

6. <u>Compute</u>

The compute key transfers control to the start of the main program. After computation has been completed and the correct answer is stored in RAM register fifteen, control is returned to CHECKNOSTROBE which will display the result while waiting for further input.

7. <u>Clear</u>

When the clear key is depressed, this code transfers control to OUTCLEAR which blanks out RAM register fifteen and then returns control to CHECKNOSTROBE.

Two service routines are used by CHECKNOSTROBE and CHECKSTROBE. They are:

1. <u>DELAY</u>

The DELAY routine is a double nested loop which, due to instruction cycle time, results in a five hundred microsecond wait.

2. <u>CHECK</u>

CHECK is a service routine which, during each transfer of a number from RAM register fifteen to another register, checks to see if the number being transferred is a fifteen. If it is, the fifteen is not transferred and control is returned to where the subroutine CHECK was called.

D. MAIN PROGRAM

The main program is the problem flow controller for the HGWI. This section of software calls the available subroutines to perform necessary arithmetic operations and character manipulations in order to arrive at a solution. The main program is entered with the input parameters loaded into their respective save registers of RAM. Register five contains the pressure altitude, register six holds the wind speed and register ten contains the scaled temperature. Registers four and nine are temporary locations used to save TI, a temperature increment used to adjust the base temperature line within a given range, and the various stages of computation of gross weight, respectively. The CPU registers are cleared and then initialized with constants which enable subsequent subroutines to compute the input temperature range. The column number, the number of rows per column, and the base address of Table I will permit computation of the address of entry into Table I. As was mentioned, the no wind gross weight for this application is determining by first computing a weight value by assuming the input parameters, temperature and pressure altitude, will fall in region II of Figure 3. The algorithm used in the solution of this application consists of the following steps:

1. First determine an initial value for W which satisfies the equation:

$$W = A(1-B \cdot TI)PA + C + D \cdot TI$$

Where A is the slope of the segment representing the highest

temperature in region II, B is the difference between the slopes of the bounding segments divided by the difference in temperature representation of the bounding segments, and C is the value of the intercept along the abcissa of the base segment within a particular region. The value D is the difference between the intercepts of the bounding segments of a particular region divided by the difference in temperature representation of those bounding segments.

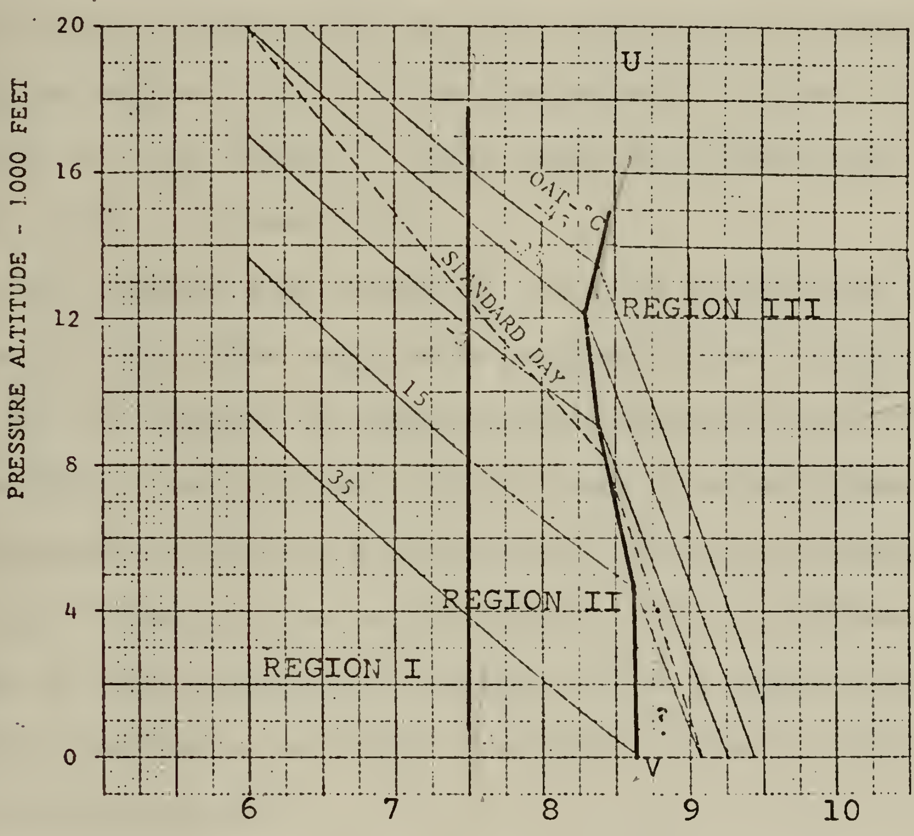



FIGURE 3 GROSS WEIGHT - 1000 LBS.

TI is the difference between the value of temperature represented by the base segment and the scaled input temperature, and PA is the input pressure altitude. The constants A, B, C and D are available in Table I. They are stored with at most three characters of significance.

2. The next step is to determine a value for $W_t$ which satisfies the equation:

$$W_t = E \cdot PA + F$$

Where E is the slope of UV, the line which separates region II from region III, and F is the intercept along the abcissa of UV.

3. The computed value of W is then compared to the computed value of $W_t$.

If W is greater than $W_t$, then the input parameters must fall in region III. If the comparison of step 3 determines that W is less than $W_t$, then step 3a is executed. Otherwise, step 4 is executed.

3a. Compare the computed value of W with the weight value, 7500, which separates region I from region II.

If the compare of step 3a determines that W is greater than 7500, then W must be the final no-wind gross weight. If, however, W is less than 7500, step 4a is executed.

4. Finally, the new constants A3, B3, C3 and D3 applicable to the segments of region III are substituted for the initial values in equation one and a final no-wind gross weight is computed.

4a. Upon reaching this final step, new constants A1, B1, C1, and D1 are substituted into equation one. W, the final no-wind gross weight, is computed.

For the HGWI/HH-IK, the final no-wind gross weight computed above must be adjusted to compensate for the scale difference between the units along the ordinate and abscissa

of Figure 3. The main program beginning at the label XINIT accomplishes this adjustment with the equation:

$$W = W/4 + 6000$$

If a flag had been set to indicate the presence of wind, the no-wind gross weight is now modified to account for the contribution to lifting capability due to wind. The correction is made using the equation:

$$W = K \cdot AS + W_o$$

Where K is the scale factor, 50, times the slope of the segment of interest, AS is the magnitude of the input wind speeds, and $W_o$ is the previously computed no-wind gross weight.

The column number for entry into Table II is determined by inspecting $W_o$. For this application, the slope of the weight correction lines is changed at each ten knot increment. Therefore, for values of wind speed in excess of ten knots, the first W computed replaces $W_o$ for the next iteration. This process, illustrated by dotted lines in Figure 4, may need to be repeated at most twice. The computed gross weight is then moved to the display register.

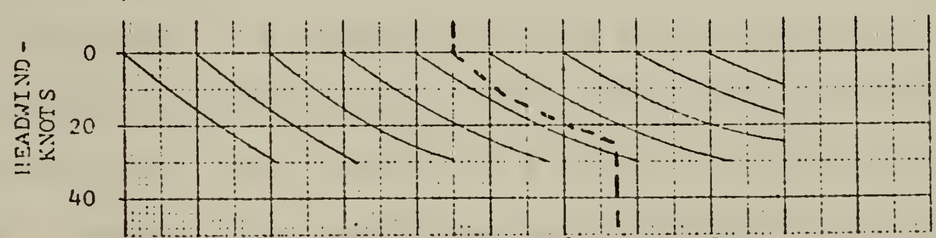



FIGURE 4 GROSS WEIGHT - 1000 LBS.

## E. SUMMARY

The data for this application was extracted from the nomograph shown in Figure 5. The information produced by this system is therefore of the same quality and value as that obtainable directly from the nomograph. The nomographs are constructed from data obtained from test flights and are adjusted to apply to a helicopter equipped with a specification engine. An adjustment is made to allow for a five per cent fuel gauge error. Manufacturers guarantee that all engines will perform at or above the level of a specification engine. The gross weight values are therefore maximum only to an aircraft with a specification engine.

Nomographs applicable to several models of operational helicopters are included in Appendix A. The similarity of the various nomographs would allow, with minor modification to this system, a general system applicable to any of the associated aircraft. The only requirement would be to replace the ROM chip which contains the tables of constants.

A system comparable to the prototype could be produced at an approximate unit cost of $350. For an associated cost increase, the system could be easily adapted to analog inputs from temperature and pressure sensing devices, requiring only that the pilot dial in the headwind speed. In this manner, one could have a continuous read-out of maximum gross weight for hovering. This would be particularly useful for the loading of troops or cargo because before the pilot lands, it would be possible to know how much added weight he could take on.

Extracted From NAVAIR 01-110HCA-1

# MAXIMUM GROSS WEIGHT FOR HOVERING

OUT OF GROUND EFFECT - MILITARY POWER
3.5°C INLET TEMPERATURE RISE
ENGINE RPM 6600

| | | | |
|---|---|---|---|
| Model(s): | UH/TH-1L ,HH-1K | Engine(s): | T53-L-13 B |
| Data as of: | August 1969 | Fuel Grade: | JP-4/JP-5 |
| DATA BASIS: | Army Phase D | Fuel Density: | 6.5 Lb/Gal. |

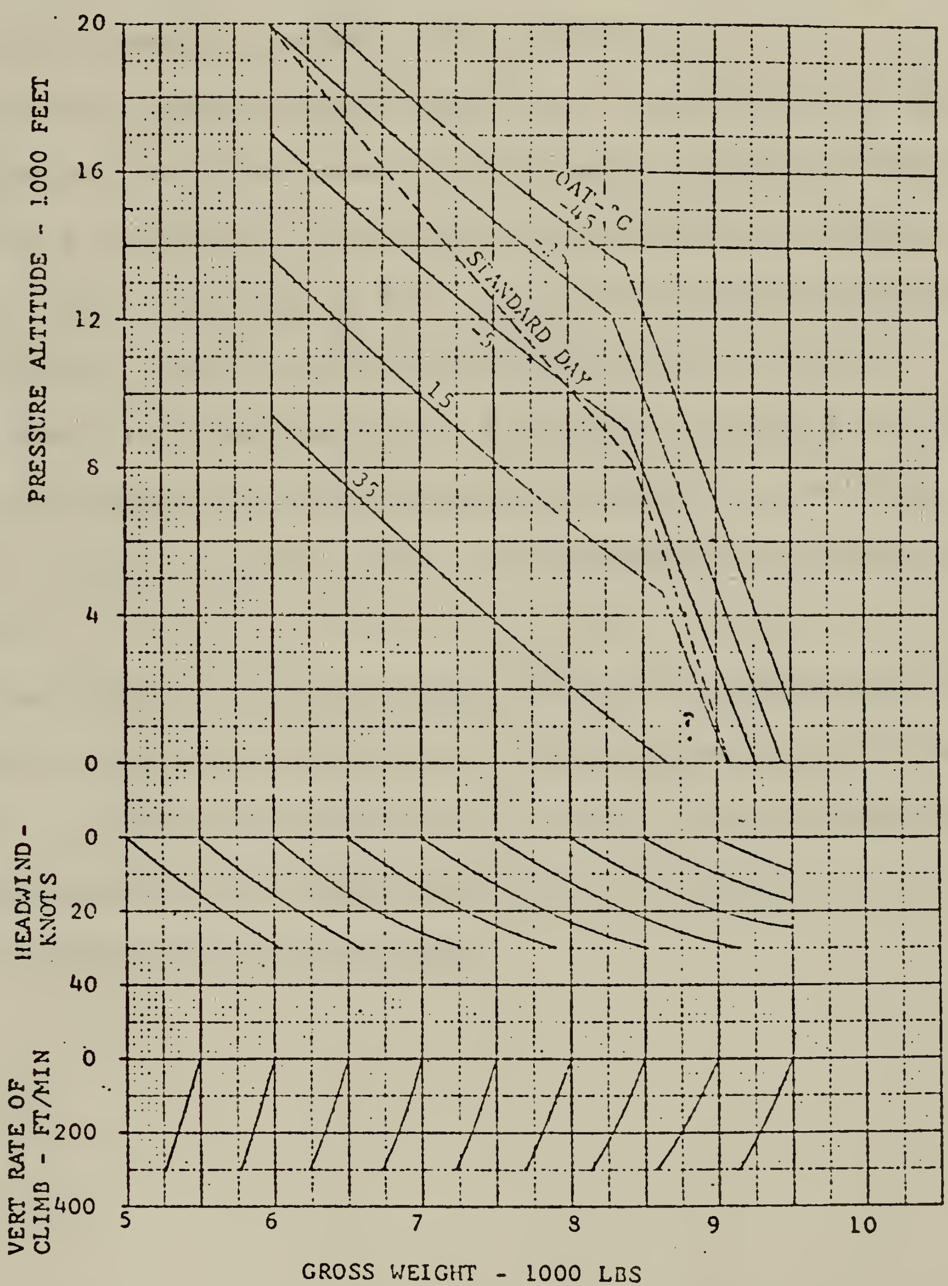



FIGURE 5.

If it were desired to allow the pilot to vary any or all of the inputs, it would be possible to keep the analog input, but also be able to set the variable at any desired value. This would allow a pilot to determine his safety margin before every take off and landing.

The primary contribution made by a system similar to the prototype described here is the rapid on-board availability of a measure of the hovering capability of the aircraft. The validity of this measure could be improved with a careful analysis and possible modification of existing data representations. Also, due to the fact that the micro-computer is programmable, alternative methods of solution could be tested until an acceptable technique is developed.

An operational system of this design is applicable to an aircraft model. A system which will produce performance data for a particular aircraft of a model is more desirable but significantly more expensive both for initial installation and subsequent maintenance.

# IV. CONCLUSION

A low cost, accurate, effective prototype of a microcomputer-based helicopter flight performance system has been built. Improvements are needed in keyboard design, data input and data storage techniques, but the feasibility of making a helicopter flight performance system work with a microcomputer has been proven. Prior to this, it was known that a man, given the time, could perform the task of figuring out maximum gross weight for hovering. It was also known that a sophisticated, but expensive, computer system could be built to accomplish basically the same task as the pilot except a computer would prove more accurate and faster than an interpolated answer. Now it is known that it is possible to build a less expensive, less sophisticated computer system. This computer system would rely on the pilot to input the desired parameters, but would produce an accurate result for a fraction of the cost of the more sophisticated system.

## APPENDIX A

Prototype operation requires depressing a sequence of number keys followed by a control key until all desired input parameter values are loaded. Parameter values for the HGWI/HH-IK must be within the ranges given below. The compute key may be depressed at any time. When the compute key is depressed, the values which are currently in the machine are used for the computation of maximum gross weight for hover.

To clear the values of temperature, altitude and wind speed, one merely depresses the corresponding control key. If one only desires to enter or change one number, all that is necessary is to enter the desired number and then depress the correct key. This will place the new number in the desired location and leave the other two inputs the same as before. The same procedure is followed if more than one number is to be changed.

There are certain ranges for acceptable input values. These are due to the ranges on the graphs and also limitations of the aircraft. If an input number is outside of these ranges, an error will be indicated by all nines appearing on the display. The acceptable ranges are:

| | |
|---|---|
| TEMPERATURE: | $-45^{\circ}$ C. to $35^{\circ}$ C. |
| WINDSPEED: | 0 Knots to 30 Knots |
| ALTITUDE: | 0 Feet to 20,000 Feet |

If an error is encountered, one merely depresses the EM

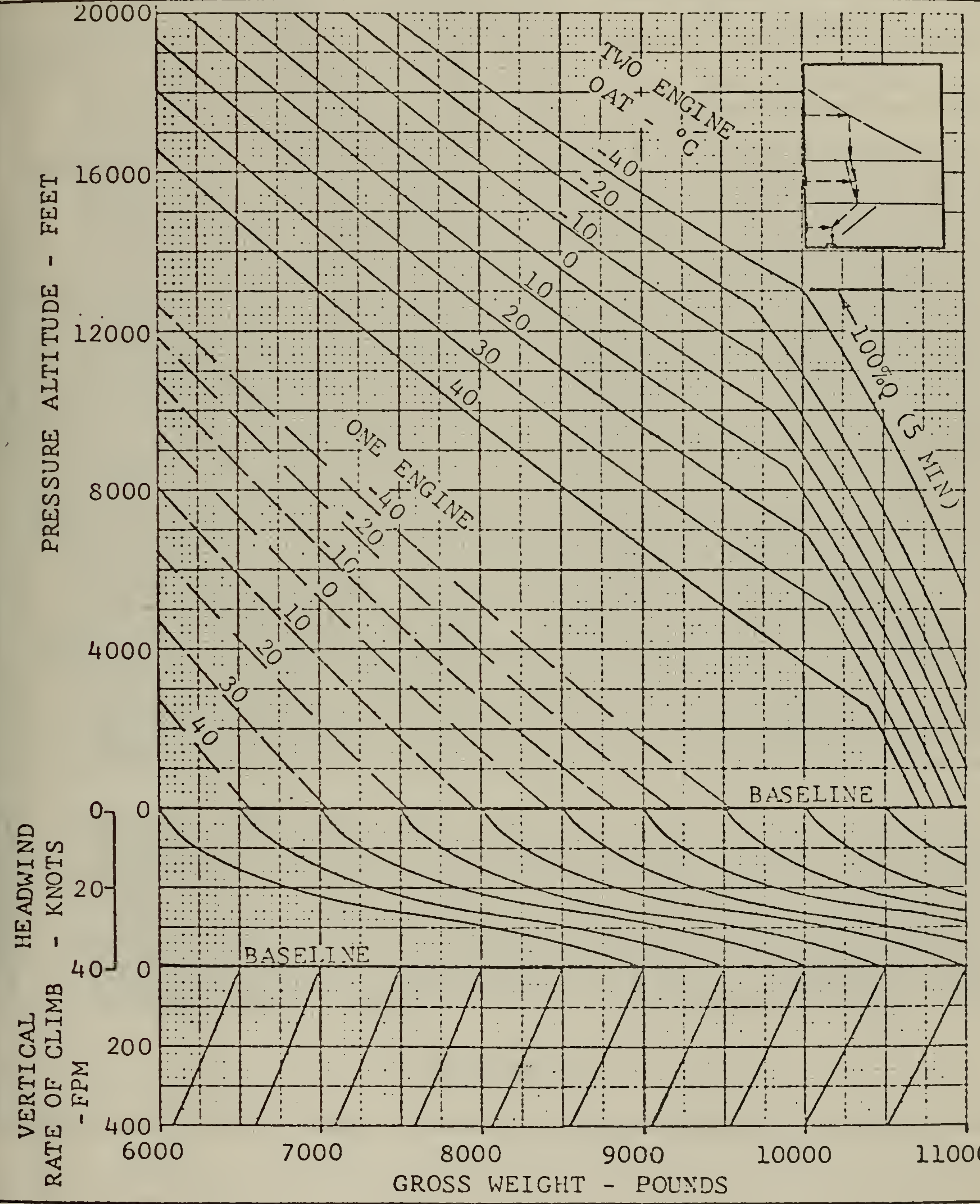
(clear) key and then inputs the correct number. If no error is encountered, the maximum gross weight for hovering will appear in the display after the compute key has been depressed. The pilot must take this number which is displayed and compare it to his present gross weight. From this comparison, the pilot decides whether a landing or take off can be made.

It is possible to recompute the maximum gross weight for hovering with different inputs with this prototype. The pilot merely depresses the clear key to clear the display and then enters the values of pressure altitude, temperature, and/or wind speed he wishes to change. After these are entered, he depresses the compute key and the new gross weight is displayed.

# HOVER CEILINGS (OUT OF GROUND EFFECT) - MRP

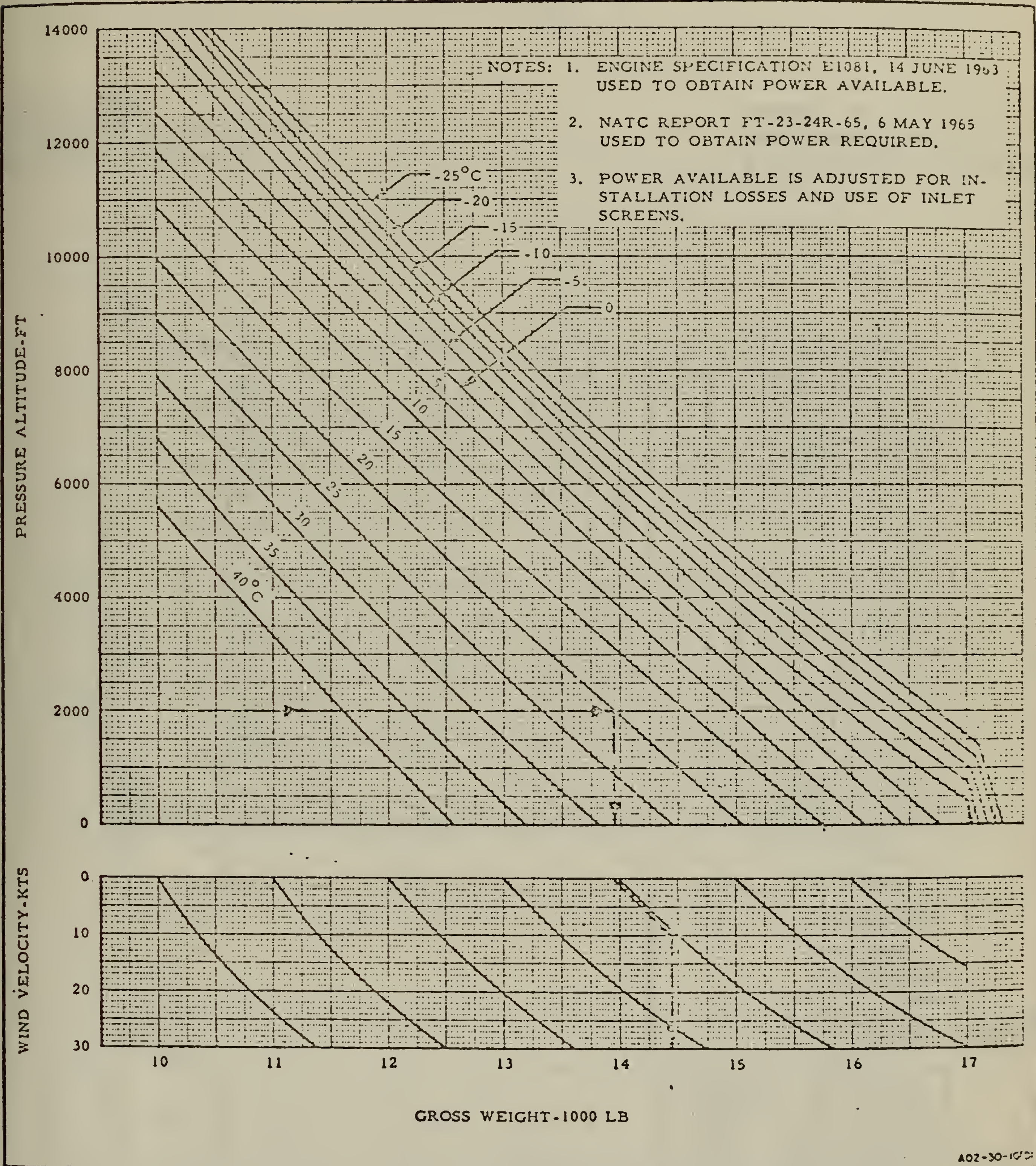
Model(s): UH-1N
Data as of: January, 1971
DATA BASIS: Bell Model 212 Flight Test

Engine(s): T400-CP-400
Fuel Grade: JP-4/JP-5
Fuel Density: 6.5 Lb/Gal



MODEL: H-46D
ENGINE(s): (1)T58-GE-10
DATA BASIS: ESTIMATED

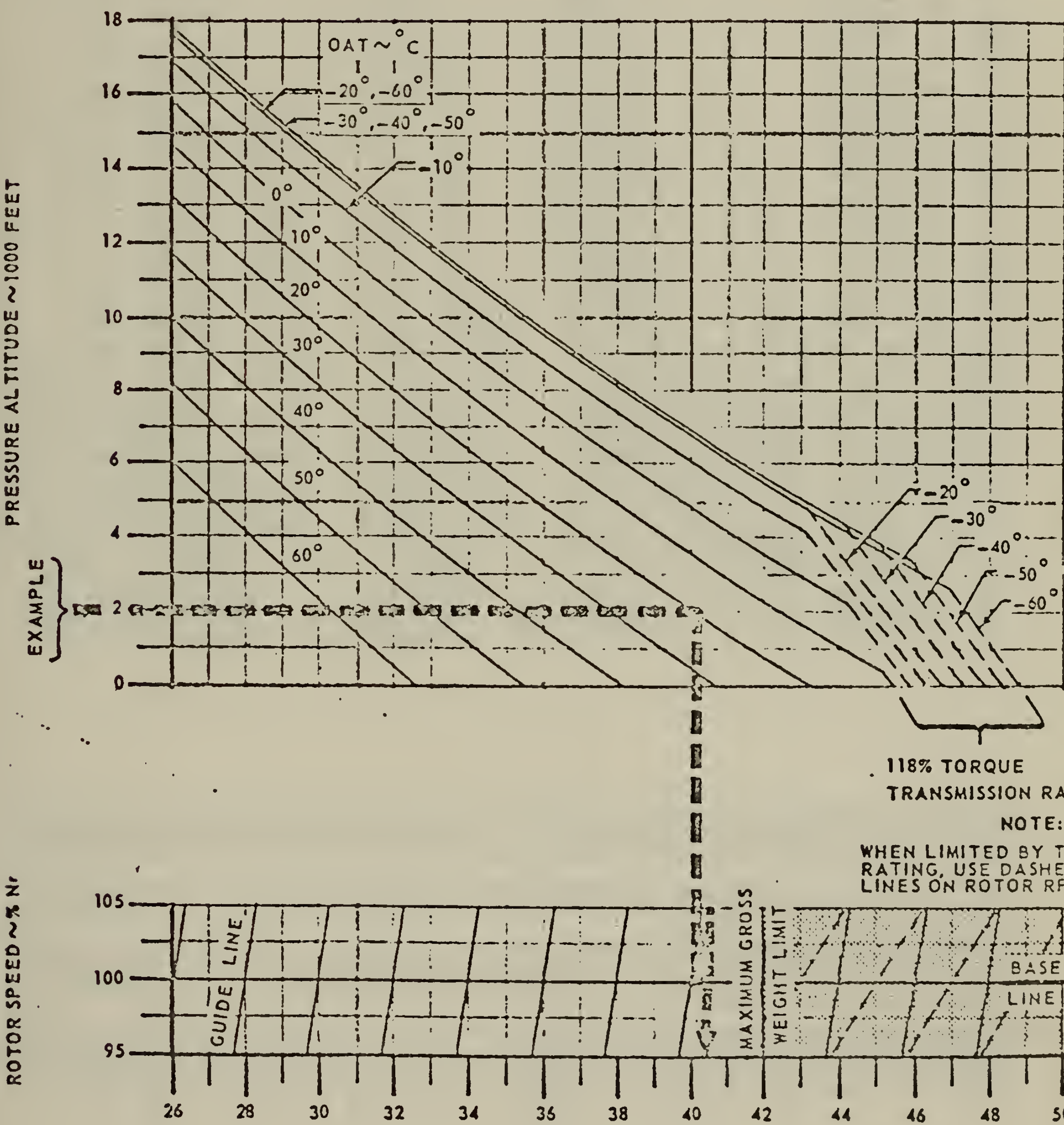
CONFIGURATION: CLEAN
ROTOR RPM: 100 PERCENT
FUEL GRADE: JP-4/JP-5



# ABILITY TO HOVER OUT OF GROUND EFFECT
# AT VARIOUS ROTOR SPEEDS
# MILITARY POWER
# 699°C $T_5$

MODEL: CH-53D
DATA AS OF: 1 NOVEMBER 1971
DATA BASIS: FLIGHT TEST

ENGINES: (2) T64-GE-413
FUEL GRADE: JP-4/JP-5
FUEL DENSITY 6.5/6.8 LB/GAL




S 3144 (R4)

```
********************
* COMPUTER PROGRAM *
********************

* MCS-4 ASSEMBLY *
0001 0000 00 000        HELICOPTER: NOP;
0002 0001 00 001        REM THESE ROUTINES MANIPULATE 16 DIGIT NUMBERS
0003 0001 00 001        IF ONE DESIRES TO USE LESS DIGITS THEN HE
0004 0001 00 001        MUST CHANGE CONSTANTS IN THE ARITHMETIC ROUTINES;
0005 0001 00 001        JUN CUTCLEAR;
0007 0003 00 003        REM SUBROUTINE MOVE MOVES NUMBERS BETWEEN RAM REGISTERS
0008 0003 00 003        IT ASSUMES THE RAM ADDRESS TO BE MOVED IS IN CPU REG AB
0009 0003 00 003        AND THE RAM REGISTER YOU WANT THE NUMBER MOVED TO IS IN
0010 0003 00 003        CPU REG EF;
0011 0003 00 003        MOVE: SRC RA; RDM; SRC RE; WRM;
0012 0007 00 007              INC RF; ISZ RB MOVE;
0013 0010 00 00A              SRC RA; RD0; SRC RE; WR0;
0014 0014 00 00E        BBL 0;
0015 0015 00 00F
0016 0015 00 00F        REM SUBROUTINE COMPLEMENT TAKES THE TENS COMPLEMENT
0017 0015 00 00F        OF A NUMBER WHICH IS IN A RAM REG. THIS REG ADDRESS
0018 0015 00 00F        MUST BE IN CPU REG EF BEFORE COMPLEMENT IS CALLED;
0019 0015 00 00F        COMPLEMENT:CLC;
0020 0016 00 010        COMPLE: LDM 10; SRC RE; SBM; CLC;
0021 0020 00 014              DAA; CMC; WRM;
0022 0023 00 017              ISZ RF COMPLE;
0023 0025 00 019        BBL 0;
0024 0026 00 01A
0025 0026 00 01A        REM SUBROUTINE ADDRS COMPUTES THE ADDRESS OF ENTRY
0026 0026 00 01A        INTO THE TABLE OF CONSTANTS AND LEAVES THIS ADDRESS
0027 0026 00 01A        IN  CPU REGISTER PAIR 0 1;
0028 0026 00 01A        ADDRS: LD R8; DAC; JCN 4 SWITCHA; XCH R8; CLC;
0029 0032 00 020             LD R7; ADD R1; XCH R1;
0030 0035 00 023           JCN 10 HOP;
0031 0037 00 025             INC R0; CLC;
0032 0039 00 027        HOP: LD R7; ADD R1; XCH R1;
0033 0042 00 02A           JCN 10 ADDRS;
0034 0044 00 02C           INC R0; CLC; JUN ADDRS;
0035 0048 00 030        SWITCHA: CLC; BBL 0;
0036 0050 00 032
0037 0050 00 032        REM SUBROUTINE SETUP IS CALLED BY MULT TO SETUP THE
0038 0050 00 032        SIGNS AND TO GET THE CORRECT NUMBERS IN THE CORRECT
0039 0050 00 032        RAM REGISTERS MULT TAKES CARE OF SETTING UP THE
0040 0050 00 032        INITIAL VALUES OF THE CPU REGS;
0041 0050 00 032        SETUP:SRC RA;  RD0; CLC; RAR; FIM RE "00";
0042 0056 00 038             JCN 2 COMPY;
0043 0058 00 03A        BACK: SRC RC; RD0; CLC; RAR; FIM RE "10";
0044 0064 00 040            JCN 2 COMPY2; JMS INIT; JUN TOPIT;
```

```
0045 0070 00 046          COMPY: JMS COMPLEMENT; JUN BACK;
0046 0074 00 04A          COMPY2: JMS COMPLEMENT; JMS INIT;
0047 0078 00 04E          TOPIT: SRC RA; RD0; XCH R5; SRC RC; RD0;
0048 0083 00 053                 ADD R5; SRC RE; WR0; LDM 0;
0049 0087 00 057                 SRC RA; WR0; SRC RC; WR0; XCH R5;
0050 0092 00 05C          BBL 0;
0051 0093 00 05D
0052 0093 00 05D          REM SUBROUTINE SHIFLT SHIFTS THE DESIGNATED RAM
                            REGISTER
0053 0093 00 05D          LEFT A DESIGNATED NUMBER OF PLACES.  THE NUMBER OF
0054 0093 00 05D          PLACES TO BE SHIFTED  IS PLACED IN CPU REGISTER PAIR C
                            C THE
0055 0093 00 05D          RAM REGISTER TO BE SHIFTED SHOULD BE PLACED IN CPU REG
                            PAIR E F;
0056 0093 00 05D          SHIFLT: NOP;
0057 0094 00 05E                  SRC RE; RDM; SRC RC; WRM; INC RD;
0058 0099 00 063                 ISZ RF SHIFLT;
0059 0101 00 065          FILLIT: WRM; ISZ RD FILLIT; CLC;
0060 0105 00 069          BBL 0;
0061 0106 00 06A
0062 0106 00 06A          REM SUBROUTINE MULT TAKES TWO  SIX  DIGIT NUMBERS,
                            POSITIVE
0063 0106 00 06A          OR NEGATIVE MULTIPLIES THEM TOGETHER AND PLACES THE
0064 0106 00 06A          ANSWER WITH THE CORRECT SIGN IN RAM REG ZERO. THE
0065 0106 00 06A          NUMBERS TO BE MULTIPLIED ARE ASSUMED TO BE IN RAM REGS
                            0 AND 1;
0066 0106 00 06A          MULT: FIM RE "20"; JMS ZERCRR; JMS INIT; JMS SETUP;
0067 0114 00 072          MULTSTART:   LD R2; XCH RB;  SRC RA; RDM;
0068 0118 00 076                      XCH R6; LD R6; JCN 4 DETOUR;
0069 0122 00 07A          MULTBEGIN: LD R3; XCH RD; SRC RC; RDM;
0070 0126 00 07E                      XCH R7; LD R2; ADD R3; CLC; XCH RF; SRC RE;
0071 0132 00 084                      LD R7; ADM; DAA; WRM; JCN 10 AROUND;
0072 0138 00 08A          TIPPY:LDM 10; XCH R4;
0073 0140 00 08C          TIP: INC RF; TCC; SRC RE; ADM; DAA; WRM;
0074 0146 00 092              ISZ R4 TIP;
0075 0148 00 094              LD RF; DAC; DAC; DAC; DAC; DAC; DAC; XCH RF;
0076 0156 00 09C          AROUND: CLC; LDM 5; SUB R3; CLC; INC R3; JCN 4
                            FINISHED;
0077 0163 00 0A3              JUN MULTBEGIN;
0078 0165 00 0A5          FINISHED: LDM 0; XCH R3; LD R6; DAC; JCN 4 DETOUR; XCH
                            R6;
0079 0172 00 0AC          ENDING: CLC; LDM 6; SUB R2; CLC; JCN 4 FINI; JUN
                            MULTBEGIN;
0080 0180 00 0B4          DETOUR:  LDM 0; XCH R3; INC R2;
0081 0183 00 0B7                  CLC; LDM 6; SUB R2; CLC; JCN 4 FINI; JUN
                            MULTSTART;
0082 0191 00 0BF          FINI:NOP;
0083 0192 00 0C0              SRC RE; RD0; CLC;
```

SHIFLT

MULT

```
0084 0195 00 0C3            RAR; JCN 2 AHEAD; JUN HEAP;
0085 02C0 00 0C8        AHEAD: FIM RE "20"; JMS COMPLEMENT;
0086 02C4 00 0CC        HEAP: FIM RA "20";
0087 02C6 00 0CE            FIM RE "0C";
0088 02C8 00 0D0          JMS ZERORR; JMS MOVE;
0089 0212 00 0D4        BBL 0;
0090 0213 00 0D5
0091 0213 00 0D5        REM SUBROUTINE ZERORR ZEROES OUT THE DESIGNATED
0092 0213 00 0D5        REGISTER.  THE REGISTER TO BE  ZEROED SHOULD BE PLACED
0093 0213 00 0D5        IN CPU REGISTER PAIR E F;
0094 0213 00 0D5        ZERORR: SRC RE; LDM 0; WRM;
0095 0216 00 0D8              ISZ RF ZERORR; WRC;
0096 0219 00 0DB        BBL 0;
0097 0220 00 0DC
0098 0220 00 0DC        REM SUBROUTINE SHIFRT SHIFTS A DESIGNATED RAM REGISTER
                          TO
0099 0220 00 0DC        THE RIGHT A DESIGNATED NUMBER OF PLACES. THE REGISTER
0100 0220 00 0DC        TO BE SHIFTED SHOULD BE PLACED IN CPU REGISTERS E AND
0101 0220 00 0DC        F.  THE NUMBER 15 SHOULD BE PLACED IN CPU REGISTER D
0102 0220 00 0DC        AND 15 MINUS THE NUMBER OF PLACES TO BE SHIFTED IN
                          REGISTER F;
0103 0220 00 0DC        SHIFRT: SRC RE; RDM; SRC RC; WRM;
0104 0224 00 0E0              LD RF; DAC; JCN 10 FILLUP;
0105 0228 00 0E4              XCH RF; LD RD; DAC; XCH RD;
0106 0232 00 0E8           JUN SHIFRT;
0107 0234 00 0EA        FILLUP: CLC; LD RD; DAC; JCN 10 OUT;
0108 0239 00 0EF              XCH RD; LDM 0; SRC RC; WRM;
0109 0243 00 0F3           JUN FILLUP;
0110 0245 00 0F5        OUT: CLC;
0111 0246 00 0F6        BBL 0;
0112 0247 00 0F7
0113 0247 00 0F7        BEGIN 260;
0114 0260 01 104
0115 0260 01 104        REM SUBROUTINE SWITCH TAKES NUMBERS OUT OF ROM AND
                          PUTS THEM
0116 0260 01 104        INTO RAM REG. IF THE NUMBER IS NEGATIVE IT PLACES THE
                          TENS
0117 0260 01 104        COMPLEMENT IN RAM, THE ASSUMPTIONS OF SWITCH ARE THAT
                          THE
0118 0260 01 104        ADDRESS OF THE ROM CONSTANT TO BE MOVED IS IN CPU REG
                          PR 0
0119 0260 01 104        AND THAT THE RAM REG YOU WANT THE NUMBER IN IS IN CPU
                          REG
0120 0260 01 104        PAIR AB;
0121 0260 01 104        SWITCH: LDM 14; XCH RF;
0122 0262 01 106        LOOP: JMS LOAD; XCH R2; SRC RA; WRM; INC RB;  XCH R3;
                          SRC RA; WRM;
0123 0271 01 10F            INC RB;
```

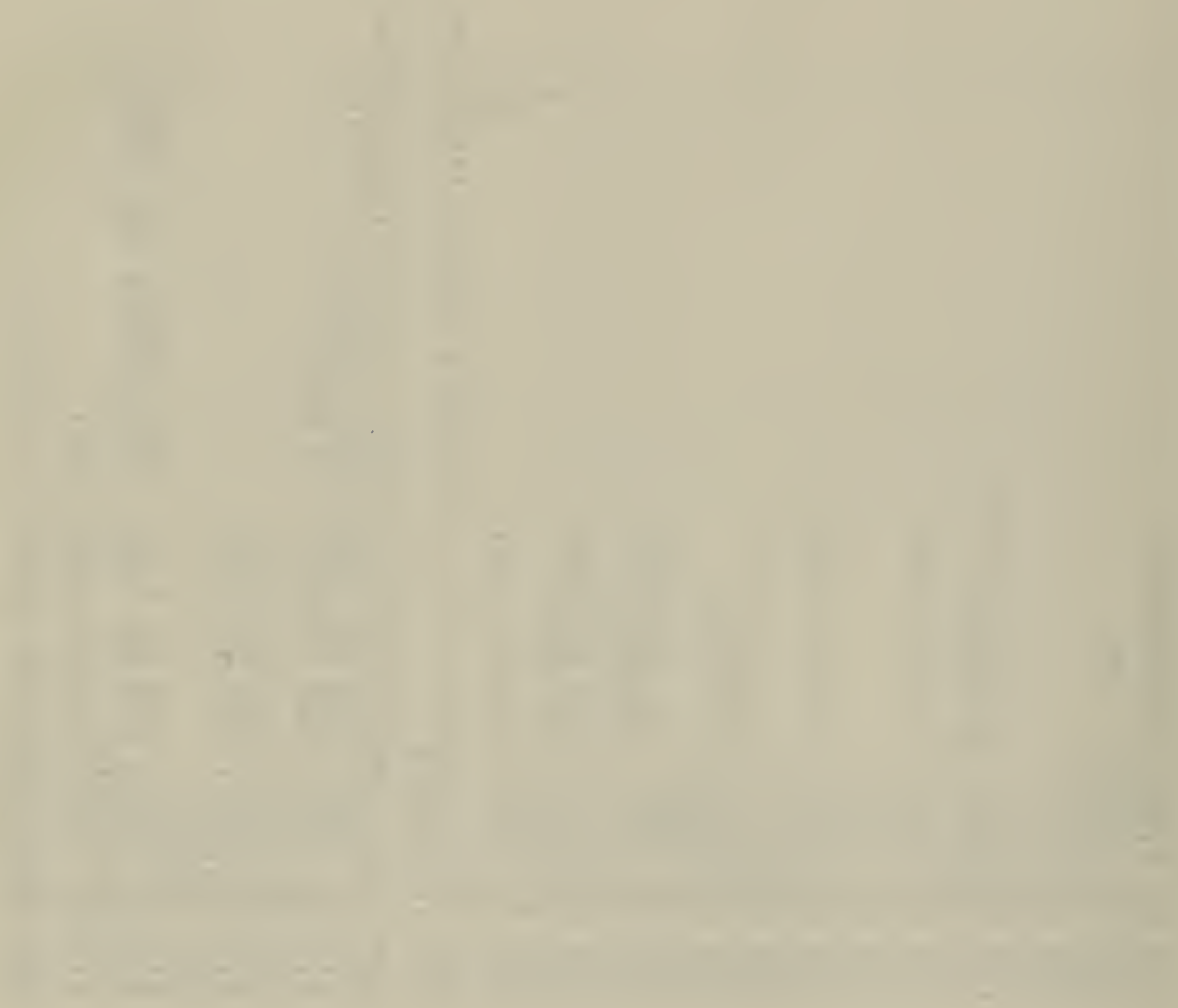
```
0272 01 110          ISZ R1 NCI1; INC RC; NCI1:
0275 01 113          ISZ RF LCCP;
0277 01 115          LD RB; DAC; XCH RB; SRC RA; RCM; WRO; LDM 0;
0284 01 11C          WRM; RDO;
C286 01 11E          CLC; RAR; JCN 2 BCTTCM;
0290 01 122      CLEAR: LDM 3; XCH RB;
0292 01 124      CLEARIT: LDM 0; SRC RA; WRM;
C295 01 127              ISZ RB CLEARIT;
0297 01 129              JUN DOWN;
C299 01 12B      BOTTOM: LDM 3; XCH RB;
0301 01 12D      NEGIT: LDM 9; SRC RA; WRM;
03C4 01 130            ISZ RB NEGIT;
03C6 01 132      DOWN:BBL 0;
03C7 01 133
C3C7 01 133      REM SUBROUTINE SUBTRACT SUBTRACTS THE NUMBER IN RAM
                   REGISTER
03C7 01 133      ZERO FROM THE NUMBER IN RAM REGISTER ONE.  THE
0307 01 133      ANSWER IN THE CORRECT FORM WILL BE PLACED IN RAM
                   RECISTER
0307 01 133      ZERO;
03C7 01 133      SUBTRACT: JMS INIT; SRC RA; RDO; CLC; RAR;
0313 01 139                JCN 2 TCADD;
0315 01 13B      SBRT: LDM 10; XCH R6;
0317 01 13D          LDM 0; XCH R7;
0319 01 13F          CLB;
0320 01 140      SUBT: SRC RC; RCM; SRC RA; SBM; JCN 2 OK;
0326 01 146          ADD R6; CLC;
0328 01 148      OK: CMC; SRC RA; WRM; INC RB; INC RD;  ISZ R7 SUBT;
0335 01 14F         JUN SENDINGS;
C337 01 151      TOADD: LDM 0; WRO; FIM RE "00"; JMS CCMPLEMENT;
0343 01 157            JMS ADDIT;
C345 01 159      SENDINGS: LDM 15; XCH RB; SRC RA; RDM; WRC; CLC;
0351 01 15F      BBL 0;
0352 01 160
C352 01 160      REM SUBROUTINE ADDIT ASSUMES TWO NUMBERS ARE IN RAM RE
                   O AND 1
0352 01 160      IT TAKES THE NUMBER IN REG 0 AND ADDS IT TO THE NUMBER
                   IN REG 1
0352 01 160      THE RESULT IS PLACED IN RAM REG 0;
0352 01 160      ADDIT: JMS INIT;
C354 01 162            LDM 0; XCH R7;
0356 01 164      ADDITION: CLC;
0357 01 165      ADDSTART: LD R2; XCH RB;
0359 01 167                LD R2; XCH RD;
0361 01 169         SRC RA; RCM;
0363 01 16B         SRC RC; ACM; DAA;
C366 01 16E         SRC RA; WRM;
0368 01 170         INC R2;
```

```
0369 01 171          ISZ R7 ADDSTART;
0371 01 173          LDM 15; XCH RB; SRC RA; RDM; WRO; CLC;
0377 01 179       DONE: BBL 0;
0378 01 17A
0378 01 17A       REM SUBROUTINE ASCOL LOOKS AT THE THOUSANDS COLUMM AND
                    THE
0378 01 17A       HUNDREDS COLUMM OF THE NC WIND GROSS WEIGHT AND SETS UP
0378 01 17A       THE CPU FOR COLNUMAS;
0378 01 17A       ASCOL: LDM 4; SRC R4; SBM;
0381 01 17D            JCN 2 NEXT; INC R8; JUN NEXT;
0386 01 182       NEXT: INC R5; SRC R4; RDM;
0389 01 185           XCH R9; CLC; BBL 0;
0392 01 188
0392 01 188       REM SUBROUTINE COLNUMAS DETERMINES THE CORRECT AIRSPEED
0392 01 188       COLUMM TO ENTER THE DATA WITH;
0392 01 188       COLNUMAS: LDM 2; XCH R6;
0394 01 18A                 CLB; XCH R7;
0396 01 18C       LOOPS: LD R8; DAC; JCN 1C COLVALA;
0400 01 190            XCH R8; LD R6; CLC; ADD R7;
0404 01 194            XCH R7; CLB; JUN LOOPS;
0408 01 198       COLVALA: LD R7; IAC; ADD RB; XCH R8;
0412 01 19C                CLC; BBL 0;
0414 01 19E
0414 01 19E       REM SUBROUTINE TESTPA DETERMINES WHETHER THE ENTERED
                    ALTITUDE
0414 01 19E       IS IN THE RANGE 0-20,000 FEET;
0414 01 19E       TESTPA: FIM R8 "52"; LDM 2; SRC R8;
0418 01 1A2               SBM; JCN 2 BYIT; JUN ERROR; BYIT: CLC; BBL 0;
0425 01 1A9       REM SUBROUTINE INITC LOADS THE PROPER CONSTANTS INTO
                    THE CPU BEFORE
0425 01 1A9       THE GROSS WEIGHT IS CALCULATED;
0425 01 1A9       INITC: JMS LOAD;
0427 01 1AB            LD R2;
0428 01 1AC            XCH R4; LD R3; XCH R5;
0431 01 1AF          ISZ R1 NOINC; INC R0;
0434 01 1B2       NOINC: JMS LOAD;
0436 01 1B4            LD R2; XCH R6; LD R3; XCH R7;
0440 01 1B8          ISZ R1 NOINC1; INC R0;
0443 01 1BB       NOINC1: JMS LOAD;
0445 01 1BD             LD R2; XCH R8; LD R3; XCH RA;
0449 01 1C1          ISZ R1 NOINC2; INC R0;
0452 01 1C4       NOINC2: BBL 0;
0453 01 1C5
0453 01 1C5       REM SUBROUTINE  TEMPCOL SETS UP THE CPU WITH THE
0453 01 1C5       PROPER NUMBERS FOR ENTRY INTO COLNUM;
0453 01 1C5       TEMPCOL: SRC R4; RDM; JCN 4 UPPER; JUN ERROR;
0459 01 1CB       REM DETERMINE COLUMN OF TABLE;
0459 01 1CB       UPPER: CLB; XCH R5; SRC R4; RDM;
```

```
0213 0463 01 1CF          INC R5; JCN 4  TENS; REM IF ACC N.E. 0 THEN
                            CONTINUE;
0214 0466 01 1D2              CLB; IAC; XCH R9;
0215 0469 01 1D5          TENS: SRC R4; RDM; REM PUT CHAR TO BE CHECKED IN R9 ;
0216 0471 01 1D7              ADD R9; XCH R9; CLC; BBL 0;
0217 0475 01 1DB
0218 0475 01 1DB          REM SUBROUTINE COLNUM DETERMINES THE PROPER COLUMM
                            NUMBER
0219 0475 01 1DB          TO ENTER THE TEMPERATURE DATA WITH;
0220 0475 01 1DB          COLNUM: CLC;  LD R6; ADD RA; XCH R6;
0221 0479 01 1DF                LD R6; SUB R9; JCN 2 COLVAL;  REM IF THE BORROW
                            IS NOT
0222 0483 01 1E3                USED THEN COLUMN NUMBER IS IN R8;
0223 0483 01 1E3             ISZ R8 COLNUM; REM IF COUNTER STILL IN RANGE
                            CONTINUE;
0224 0485 01 1E5             JUN ERROR;
0225 0487 01 1E7          COLVAL: CLC; LD R8; REM SET R8 TO COLUMN NUMBER AND
                            RETURN;
0226 0489 01 1E9                SUB R7; XCH R8;
0227 0491 01 1EB          BBL 0;
0228 0492 01 1EC
0229 0492 01 1EC          REM SUBROUTINE INIT CLEARS THE ACCUMULATOR AND THE
                            CARRY
0230 0492 01 1EC          THEN PUTS ZEROES INTO CPU REG 2 3 4 5 6 7 8 9 AND A
                            '00'
0231 0492 01 1EC          IN REG PAIR AB AND A '10' IN REG PAIR CD AND A '20' IN
                            REG PAIR EF;
0232 0492 01 1EC          INIT: CLB;
0233 0493 01 1ED               FIM R2 "00";
0234 0495 01 1EF               FIM R4 "00";
0235 0497 01 1F1               FIM R6 "00";
0236 0499 01 1F3               FIM R8 "00";
0237 0501 01 1F5               FIM RA "00";
0238 0503 01 1F7               FIM RC "10";
0239 0505 01 1F9               FIM RE "20";
0240 0507 01 1FB          BBL 0;
0241 0508 01 1FC
0242 0508 01 1FC          BEGIN 512;
0243 0512 02 200
0244 0512 02 200          REM SUBROUTINE LOADROBAS INITIALIZES THE CPU REGISTERS
0245 0512 02 200          WITH CONSTANTS STORED IN FRONT OF THE CONSTANT TABLES.
0246 0512 02 200          THE CONSTANTS, TABLE BASE ADDRESS AND NUMBER OF ROWS
                            PER
0247 0512 02 200          COLUMN ARE USED TO COMPUTE TABLE ENTRY ADDRESS.  ;
0248 0512 02 200          LOADROBAS: JMS LOAD; REM PUT NO. OF ROWS/COL IN R7;
0249 0514 02 202                     LD R3; XCH R7; CLC; LDM 1;
0250 0518 02 206                     ADD R1; XCH R1; TCC; ADD R0;
0251 0522 02 20A                     XCH R0; JMS LOAD; LD R2; XCH R0;
```

```
0252 0527 02 20F                LD R3; XCH R1; CLC; BBL 0;
0253 0531 02 213
0254 0531 02 213       REM SUBROUTINE ASCHECK CHECKS THE MAGNITUDE OF THE
0255 0531 02 213       INPUT WIND SPEED TO INSURE IT IS GREATER THAN OR
0256 0531 02 213       EQUAL TO ZERRO BUT LESS THAN OR EQUAL TO THIRTY.  ;
0257 0531 02 213       ASCHECK: FIM RA "60"; FIM RC "6A";
0258 0535 02 217                SRC RA; RDM; DAC; TCC; SRC RC;
0259 0540 02 21C                WRM; INC RB; SRC RA;
0260 0543 02 21F                RDM; SRC RC; ADM; WRM;
0261 0547 02 223                LDM 3; SBM; JCN 2 BYEBYE;
0262 0551 02 227           JUN ERROR;
0263 0553 02 229       BYEBYE: RDM; WR1; CLB; WRM;
0264 0557 02 22D       BBL 0;
0265 0558 02 22E
0266 0558 02 22E       REM MAIN PROGRAM BEGINS HERE. ;
0267 0558 02 22E       START: NOP;
0268 0559 02 22F
0269 0559 02 22F       REM ZERRO OUT WORKING REGISTERS.  ;
0270 0559 02 22F       FIM RE "00"; JMS ZERORR; WR1;
0271 0564 02 234       FIM RE "10"; JMS ZERORR; WR1;
0272 0569 02 239       FIM RE "20"; JMS ZERORR; WR1;
0273 0574 02 23E       FIM RE "40"; JMS ZERORR; WR1;
0274 0579 02 243       FIM RE "7C"; JMS ZERORR; WR1;
0275 0584 02 248       FIM RE "90"; JMS ZERORR; WR1;
0276 0589 02 24D       FIM RE "A0"; SRC RE; CLB; WR1;
0277 0594 02 252       JMS INIT;
0278 0596 02 254       FIM RO "00";
0279 0598 02 256       JMS INITC;
0280 0600 02 258       JMS TEMPCOL;
0281 0602 02 25A       JMS COLNUM;
0282 0604 02 25C       JMS LOADROBAS;
0283 0606 02 25E       JMS ADDRS; REM LEAVES ROM ADDR IN RO/1;
0284 0608 02 260       FIM RA "10";
0285 0610 02 262       JMS SWITCH;
0286 0612 02 264       FIM RA "A0";
0287 0614 02 266       FIM RE "00";
0288 0616 02 268       JMS ZERORR; JMS MOVE;
0289 0620 02 26C       JMS SUBTRACT; REM LEAVES T1 IN RR-0;
0290 0622 02 26E       FIM RA "00";
0291 0624 02 270       FIM RE "40";
0292 0626 02 272       JMS ZERORR; JMS MOVE;
0293 0630 02 276
0294 0630 02 276       BEGIN 768;
0295 0768 03 300       START2: FIM RA "40"; FIM RE "00";
0296 0772 03 304          JMS ZERORR; JMS MOVE;
0297 0776 03 308       FIM RA "10";
0298 0778 03 30A       JMS SWITCH;
0299 0780 03 30C       JMS MULT;
```

```
0782 03 30E    FIM RE "10"; JMS ZERORR;
0786 03 312    FIM RA "13";
0788 03 314    SRC RA; LDM 1; WRM;
0791 03 317    JMS SUBTRACT;
0793 03 319    FIM RA "10";
0795 03 31B    JMS SWITCH;
0797 03 31D    JMS MULT;
0799 03 31F    FIM RA "50"; FIM  RE "10";
0803 03 323    JMS ZERORR; JMS MOVE;
0807 03 327    JMS MULT;
0809 03 329    FIM RC "00"; FIM RE "03";
0813 03 32D    JMS SHIFLT;
0815 03 32F    FIM RA "10";
0817 03 331    JMS SWITCH;
0819 03 333    FIM RC "1F"; FIM RE "1D";
0823 03 337    JMS SHIFRT;
0825 03 339    JMS ADDIT;
0827 03 33B    FIM  RA "00"; FIM RE  "7C";
0831 03 33F    JMS ZERORR; JMS MOVE;
0835 03 343    FIM RA "10";
0837 03 345    JMS SWITCH;
0839 03 347    FIM RA "40"; FIM RE "00";
0843 03 34B    JMS ZERORR; JMS MOVE;
0847 03 34F    JMS MULT;
0849 03 351    FIM RA "70"; FIM RE "10";
0853 03 355    JMS ZERORR; JMS MOVE;
0857 03 359    JMS ADDIT;
0859 03 35B    FIM RA "00"; SRC RA; RD1;
0863 03 35F
0863 03 35F    REM TEST FOR FLAG SET TO INDICATE SECOND ITERATION. ;
0863 03 35F    JCN 12 XINIT;
0865 03 361
0865 03 361    REM TEST COMPUTED GROSS WEIGHT FOR POSSIBLE REGION 3.
0865 03 361    FIM RA "00"; FIM RE "90"; JMS ZERORR; JMS MOVE;
0873 03 369    FIM RA "10"; JMS SWITCH; FIM RA "50";
0879 03 36F    FIM RE "00"; JMS ZERORR; JMS MOVE;
0885 03 375    JMS MULT;
0887 03 377    FIM RA "10"; JMS SWITCH;
0891 03 37B    FIM RC "1F"; FIM RE "1D";
0895 03 37F    JMS SHIFRT;
0897 03 381    JMS ADDIT;
0899 03 383    FIM RA "00"; FIM RE "10"; JMS ZERORR; JMS MOVE;
0907 03 38B    FIM RA "90"; FIM RE "00"; JMS ZERORR; JMS MOVE;
0915 03 393    JMS SUBTRACT;
0917 03 395    FIM RA "0F"; SRC RA; RDM; WR1;
0922 03 39A
0922 03 39A    REM TEST FOR GROSS WEIGHT IN REGION 1.  ;
0922 03 39A    JCN 12 START2;
```



```
0348 0924 03 39C          FIM RA "90"; FIM RE "1C"; JMS ZERORR; JMS MOVE;
0349 0932 03 3A4          FIM RE "00"; JMS ZERORR;
0350 0936 03 3A8          FIM RA "03"; SRC RA;
0351 0939 03 3AB          LDM 6; WRM; JMS SUBTRACT;
0352 0943 03 3AF          LDM 8; ADD R1; XCH R1;
0353 0946 03 3B2          TCC; ADD R0; XCH R0;
0354 0949 03 3B5          FIM RA "0F"; SRC RA; RDM; WR1;
0355 0954 03 3BA          JCN 12 START2;
0356 0956 03 3BC          FIM RA "90"; FIM RE "00"; JMS ZERORR; JMS MOVE;
0357 0964 03 3C4
0358 0964 03 3C4          REM ADJUST NO-WIND GROSS WEIGHT FOR SCALE DIFFERENCE.
0359 0964 03 3C4          XINIT:
0360 0964 03 3C4            FIM RE "10"; JMS ZERORR; FIM RE "20";
0361 0970 03 3CA            JMS ZERORR; FIM RA "10"; SRC RA;
0362 0975 03 3CF            LDM 5; WRM; INC RB; SRC RA;
0363 0979 03 3D3            LDM 2; WRM; JMS MULT;
0364 0983 03 3D7          FIM RC "00"; FIM RE "02"; JMS SHIFLT; FIM RE "10";
0365 0991 03 3DF            JMS ZERORR; FIM RA "13"; SRC RA; LDM 6;
0366 0997 03 3E5            WRM; JMS ADDIT; FIM RA "00"; FIM RE "90";
0367 1004 03 3EC            JMS MOVE;
0368 1006 03 3EE
0369 1006 03 3EE          BEGIN 1024;
0370 1024 04 400
0371 1024 04 400          REM ADJUST COMPUTED X FOR WIND;
0372 1024 04 400          FIM RA "60"; SRC RA; RD1;
0373 1028 04 404          JCN 4 ASZERO;
0374 1030 04 406          FIM R0 "05"; JMS INIT; JMS INITC;
0375 1036 04 40C          JMS ASCOL;
0376 1038 04 40E          JMS COLNUM;
0377 1040 04 410          JMS COLNUMAS;
0378 1042 04 412          JMS LOADROBAS;
0379 1044 04 414          JMS ADDRS;
0380 1046 04 416          CLC; FIM RC "61"; SRC RC; RDM;
0381 1051 04 41B          TENKTS: DAC; WR1;
0382 1053 04 41D          JCN 10 UNITS; FIM RA "10"; JMS SWITCH;
0383 1059 04 423          FIM RE "00"; JMS ZERORR; SRC RE;
0384 1064 04 428          LDM 5; WRM; JMS MULT;
0385 1068 04 42C          FIM RA "90"; FIM RE "10"; JMS ZERORR;
0386 1074 04 432          JMS MOVE; JMS ADDIT;
0387 1078 04 436          FIM RA "00"; FIM RE "90"; JMS ZERORR; JMS MOVE;
0388 1086 04 43E          FIM RA "60"; SRC RA;
0389 1089 04 441          CLB; RD1; JUN TENKTS;
0390 1093 04 445          UNITS:
0391 1093 04 445            FIM RE "00"; JMS ZERORR;
0392 1097 04 449            FIM RC "60"; SRC RC; RDM;
0393 1101 04 44D            SRC RE; WRM;
0394 1103 04 44F            FIM RE "10"; JMS ZERORR;
0395 1107 04 453            INC RF; SRC RE; LDM 5;
```

```
0396 1110 04 456          WRM; JMS MULT;
0397 1113 04 459          FIM RA "10"; JMS SWITCH;
0398 1117 04 45D       JMS MULT; FIM RC "00"; FIM RE "02"; JMS SHIFLT;
0399 1125 04 465          FIM RA "90"; FIM RE "10";
0400 1129 04 469          JMS ZERORR; JMS MOVE;
0401 1133 04 46D          JMS ADDIT;
0402 1135 04 46F       FIM RA "00"; FIM RE "9C"; JMS MOVE;
0403 1141 04 475       ASZERO: NOP;
0404 1142 04 476
0405 1142 04 476       REM MOVE COMPUTED GROSS WEIGHT TO DISPLAY REGISTER.  ;
0406 1142 04 476       FIM RA "90"; FIM RE "F0"; JMS MOVE;
0407 1148 04 47C       FIM RE "F4";
0408 1150 04 47E       FILLEMUP: SRC RE; LDM "F"; WRM;
0409 1153 04 481          ISZ RF FILLEMUP;
0410 1155 04 483       JUN CHECKNOSTROBE;
0411 1157 04 485
0412 1157 04 485       BEGIN 1280;
0413 1280 05 500
0414 1280 05 500       REM SUBROUTINE DELAY IS CALLED TO COMPHENSATE FOR
0415 1280 05 500       MECHANICAL INCONSISTANCIES OF KEYBOARD INPUT.  ;
0416 1280 05 500       DELAY: FIM R6 "00";
0417 1282 05 502       ITDELAY: ISZ R7 ITDELAY;
0418 1284 05 504                ISZ R6 ITDELAY;
0419 1286 05 506       BBL 0;
0420 1287 05 507
0421 1287 05 507       REM INPUT OUTPUT ROUTINES FOLLOW.  ;
0422 1287 05 507       CHECK: SRC RC; RDM; IAC; CLC;
0423 1291 05 5CB          JCN ZAC OVERALL;
0424 1293 05 50D             DAC; CLC; SRC RE; WRM; INC RF;
0425 1298 05 512          ISZ RC CHECK;
0426 1300 05 514       OVERALL: BBL 0;
0427 1301 05 515       CHECKSTROBE: FIM RE "F0"; JMS "600";
0428 1305 05 519       FIM RC "10"; SRC RC; RDR; RAL;
0429 1310 05 51E       JCN 10 CHECKSTROBE; JMS DELAY; FIM RC "00"; SRC RC;
                          RDR; XCH RC;
0430 1319 05 527       INC RC; SRC RC; RDR; RAL;
0431 1323 05 52B       JCN 10 CHECKSTROBE;
0432 1325 05 52D       FIM R0 KEYTAB; RAL; JCN 10 NUMBERSIDE;
0433 1330 05 532       INC R0; INC R0;
0434 1332 05 534       NUMBERSIDE: LD RD;
0435 1333 05 535       CLC; ADD RD; JCN 10 DC1; INC R0;
0436 1338 05 53A       DC1: CLC; ADD R1; XCH R1;
0437 1341 05 53D        JCN 10 DC2; INC R0;
0438 1344 05 540       DC2: LD RD; JIN R0;
0439 1346 05 542       CHECKNOSTROBE: FIM RE "FC"; JMS "600";
0440 1350 05 546        FIM RC "10"; SRC RC;
0441 1353 05 549          RDR; RAL; JCN 2 CHECKNOSTROBE;
0442 1357 05 54D        JUN CHECKSTROBE;
```

```
0443 1359 05 54F          KEYTAB: JUN CHECKNOSTROBE;
0444 1361 05 551                  JUN NUMBER; JUN NUMBER; JUN NUMBER;
0445 1367 05 557                  JUN NUMBER; JUN NUMBER; JUN NUMBER;
0446 1373 05 55D                  JUN NUMBER; JUN NUMBER; JUN NUMBER;
0447 1375 05 563                  JUN CHECKNOSTROBE; JUN CHECKNOSTROBE; JUN
                            CHECKNOSTROBE;
0448 1385 05 569                  JUN START;
0449 1387 05 56B                  JUN CHECKNOSTROBE; JUN CHECKNOSTROBE;
0450 1391 05 56F                  JUN NUMBER;
0451 1393 05 571                  JUN CHECKNOSTROBE;
0452 1395 05 573                  JUN NEGATIVE;
0453 1397 05 575                  JUN CHECKNOSTROBE; JUN CHECKNOSTROBE; JUN
                            CHECKNOSTROBE;
0454 1403 05 57B                  JUN ALTITUDE;
0455 1405 05 57D                  JUN CHECKNOSTROBE;
0456 1407 05 57F                  JUN OUTCLEAR;
0457 1409 05 581                  JUN CHECKNOSTROBE; JUN CHECKNOSTROBE; JUN
                            CHECKNOSTROBE;
0458 1415 05 587                  JUN SPEED;
0459 1417 05 589                  JUN TEMPERATURE;
0460 1419 05 58B                  JUN CHECKNOSTROBE; JUN CHECKNOSTROBE;
0461 1423 05 58F
0462 1423 05 58F          REM SCALE INPUT TEMPERATURE WITH 45 DEGREES.  ;
0463 1423 05 58F          TEMPERATURE: FIM RE "AO"; JMS ZERORR;
0464 1427 05 593          FIM RC "FO"; JMS CHECK;
0465 1431 05 597          COVEDALL: FIM RE "AO"; SRC RE; RD3; CLC; RAR;
0466 1437 05 59D             JCN 2 JUMPY; JUN SHAKY;
0467 1441 05 5A1          JUMPY: FIM RE "AO"; JMS COMPLEMENT; CLB; WR3;
0468 1447 05 5A7          SHAKY: SRC RE; CLB;
0469 1449 05 5A9          LDM 5; ADM; DAA; WRM; LDM 4; INC RF;
0470 1455 05 5AF          SRC RE; ADM; DAA; WRM; INC RF; LDM 0;
0471 1461 05 5B5          ROUND: SRC RE; ADM; DAA; WRM;
0472 1465 05 5B9             ISZ RF ROUND;
0473 1467 05 5BB          JUN OUTCLEAR;
0474 1469 05 5BD          SPEED: FIM RE "60"; JMS ZERORR; FIM RC "FO";
0475 1475 05 5C3          JMS CHECK;
0476 1477 05 5C5          UNDERALL: JMS ASCHECK; JUN OUTCLEAR;
0477 1481 05 5C9          ALTITUDE: FIM RE "50"; JMS ZERORR; FIM RC "F2";
0478 1487 05 5CF          JMS CHECK;
0479 1489 05 5D1          OVERITS: JMS TESTPA; JUN OUTCLEAR;
0480 1493 05 5D5          NUMBER: FIM RA "EO"; SRC RA; WRM; FIM RC "FF"; FIM RE
                            "FE";
0481 1501 05 5DD             JMS SHIFRT; SRC RA; RDM; FIM RA "FO"; SRC RA; WRM;
0482 1509 05 5E5             JUN CHECKNOSTROBE;
0483 1511 05 5E7          NEGATIVE: FIM RE "AO"; JMS ZERORR; LDM 1; SRC RE; WR3;
0484 1518 05 5EE             JUN CHECKNOSTROBE;
0485 1520 05 5FC          OUTCLEAR: FIM RE "FO"; CLB;
0486 1523 05 5F3                  SFFFS: SRC RE; LDM "F"; WRM;
```

```
1526 C5 5F6          ISZ RF SFFFS; JUN CHECKNOSTROBE;
1530 C5 5FA
1530 C5 5FA     BEGIN 1536;
1536 C6 600
1536 06 600     REM SUBROUTINE DISPLAY DISPLAYS THE CHARACTERS IN RAM
                  REGISTER
1536 06 600     F THROUGH THE USE OF MULTIPLEXING;
1536 06 600     DISPLAY: FIM R8 0; REM BCD PORT;
1538 06 602            FIM RA 16; REM CONTROL PORT;
1540 C6 604            LDM 0; XCH R2;
1542 06 606            FIM RC 204; REM LOOP COUNTERS;
1544 06 608            SRC RE; RD0; XCH R4; RD1;
1548 C6 60C            XCH R5; RD2; XCH R6; RD3; XCH R7;
1553 06 611     REM HANDLE DECIMAL POINT HERE;
1553 06 611     LP: LD  R2;RAL; STC; RAR;
1557 C6 615         XCH R2; REM DECIMAL BIT IS OFF (1);
1558 C6 616            LD R4; RAL; XCH R4; JCN 10 CLOK;
1563 06 61B             LD R2;
1564 C6 61C            RAL; CLC; RAR; XCH R2; REM DECIMAL BIT IS ON
                  (0);
1568 06 620     REM CLOCK SHIFT REGISTER;
1568 06 620     CLOK: CLB; LD R2; SRC RA; WRR; REM SEND DATA BEFORE
                  SHIFT;
1572 06 624     LDM 4; ADD R2; SRC RA; WRR; REM CLOCK NOW HIGH (1);
1576 C6 628     REM BLANK DISPLAY DURING SHIFT;
1576 06 628     LDM 15; SRC R8; WRR; SRC RA;
1580 06 62C     REM DROP CLOCK;
1580 06 62C     LD R2; WRR; REM SEND RAM DIGIT;
1582 06 62E     SRC  RE; RDM; SRC R8; WRR;
1586 06 632     INC RF; REM POINT TO NEXT DIGIT IN RAM;
1587 06 633     REM MAKE SURE DATA BIT IS ONE DURING REMAINING
1587 06 633     SHIFTS TO DISABLE ALL BUT ONE ANODE;
1587 06 633     LD R2; RAR; STC; RAL; XCH R2;
1592 06 638     REM WAIT AWHILE (300 USEC????);
1592 06 638     LDM 0; XCH R3; WAIT: ISZ R3 WAIT;
1596 06 63C     REM LOOP BACK TO DISPLAY REMAINING DIGITS;
1596 06 63C     ISZ RC LP; LD R5;
1599 06 63F     XCH R4; LD R6; XCH R5; LD R7;
1603 06 643     XCH R6; LDM 12; XCH RC;
1606 06 646     ISZ  RD LP; SRC RA; LDM 13; WRR;
1611 C6 64B     LDM 9; REM DP OFF, CLOCK OFF;
1612 06 64C     WRR;
1613 06 64D     BBL 0;
1614 06 64E
1614 06 64E     BEGIN 1792;
1792 07 700     CON "AF08C20F0A925CC1038202005000778114208610";
1812 07 714     CON "1300550C9991269183100400200078816220411O";
1832 07 728     CON "04007991778121206410789180100000269103100";
```

```
0532 1852 07 73C          CON "0400999158816910151C06002100598164100330";
0533 1872 07 750          CON "99916110899156912210040020009881151O5220";
0534 1892 07 764          CON "0800299197816010891C0C006010000056916010";
0535 1912 07 778          CON "080020003981101025200600080C011C070CC900";
0536 1932 07 78C          CON "021007000010031008000010051CC90001100520";
0537 1952 07 7A0          CON "0010521C000C0010CCCC0000";
0538 1964 07 7AC          ERROR: FIM RE "FO";
0539 1966 07 7AE           ERR1: SRC RE; LDM 9; WRM;
0540 1969 07 781                 ISZ RF ERR1;
0541 1971 07 7B3          JUN CHECKNOSTROBE;
0542 1973 07 7B5          LOAD: FIN R2;
0543 1974 07 7B6          L1: BBL 0;
0544 1975 07 7B7          FINY: NOP;
0545 1976 07 7B8          END
```

## V. LIST OF REFERENCES


1. U.S. ARMY AVIATION SYSTEMS TEST ACTIVITY, Engineering Flight Evaluation of Helicopter Lift Margin System Test Plan, by F. Dominick, April 1973.

2. ELLIOTT ELECTRONICS, Hermes Helicopter Energy and Rotor Management System, p. 2-17.

3. INTEL CORPORATION, MCS-4 Micro Computer Users Manual, Revision 3, July 1972.

4. NAVAL AIR SYSTEMS COMMAND NAVAIR 01-230HMA-1, NATOPS Flight Manual Navy Model CH-53 A/D Helicopters, p. 11-23, 1 November 1971.

5. NAVAL AIR SYSTEMS COMMAND NAVAIR 01-110HCE-1, NATOPS Flight Manual UH-IN Helicopter, p. 11-20, January 1971.

6. NAVAL AIR SYSTEMS COMMAND NAVAIR 01-250HDA-1, NATOPS Flight Manual H46-D Helicopter, p. 11-41, 6 May 1965.

7. NAVAL AIR SYSTEMS COMMAND NAVAIR 01 110HCA-1, NATOPS Flight Manual UH/TH-IL, HHIK Helicopters, p. 11-76, August 1969.

8. NAVAL AIR TEST CENTER USNTPS-FTM NO. 101, Naval Test Pilot School Flight Test Manual Helicopter Stability and Control, 10 June 1968.

9. RICHARDS, R.B., Principles of Helicopter Performance, Naval Air Test Center, USNTPS-T-No. 1, 8 March 1968.

INITIAL DISTRIBUTION LIST

| | | No. of Copies |
|---|---|---|
| 1. | Defense Documentation Center<br>Cameron Station<br>Alexandria, Virginia 22314 | 2 |
| 2. | Library, Code 0212<br>Naval Postgraduate School<br>Monterey, California 93940 | 2 |
| 3. | Chairman Computer Science Group, Code 72<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 4. | Professor G.A. Kildall, Code 53 Kd<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 5. | Lieutenant (j.g.) R.H. Brubaker, Jr., Code 53Bh<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 6. | Capt. E.E. Eloe<br>14521 Apt. A Carfax<br>Tustin, California 92680 | 1 |
| 7. | Lt. R.T. Scott, Jr.<br>1428 Carissa Ct.,<br>Chula Vista, California 92011 | 1 |
| 8. | Mr. G. Flohill, Code 461<br>Office of Naval Research<br>Arlington, Virginia 22217 | 1 |
| 9. | Commander<br>US Army Aviation System Test Activity<br>Attn: SAVTE-M (Mr. Farrell)<br>Edwards AFB, California 93523 | 1 |
| 10. | Major Richard L. Phillips, USMC<br>Sub Unit #1, MCTSSA<br>Marine Corps Air Station (H)<br>Santa Ana, California 92075 | 1 |

Security Classification

**DOCUMENT CONTROL DATA - R & D**

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

| ORIGINATING ACTIVITY (Corporate author) | 2a. REPORT SECURITY CLASSIFICATION |
|---|---|
| Naval Postgraduate School<br>Monterey, California 93940 | Unclassified |
| | 2b. GROUP |

REPORT TITLE

A Helicopter Flight Performance System Using an LSI Microprocessor.

DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; (June 1973)

AUTHOR(S) (First name, middle initial, last name)

Edwin Eugene Eloe and Richard Tazewell Scott, Jr.

| REPORT DATE | 7a. TOTAL NO. OF PAGES | 7b. NO. OF REFS |
|---|---|---|
| June 1973 | 59 | 9 |

| | |
|---|---|
| 8a. CONTRACT OR GRANT NO. | 9a. ORIGINATOR'S REPORT NUMBER(S) |
| b. PROJECT NO. | |
| c. | 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) |
| d. | |

DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

| SUPPLEMENTARY NOTES | 12. SPONSORING MILITARY ACTIVITY |
|---|---|
| | Naval Postgraduate School<br>Monterey, California 93940 |

ABSTRACT

This thesis presents the development of a helicopter gross weight calculator. Required qualities for aircraft system components such as durability, reliability, and low weight are met using an LSI micro-processor. The prototype system which was developed weighs approximately four pounds and has approximate dimensions of 7" x 5" x ½". The cost estimate for the system is less than $350. The calculator solution is based on the solution technique currently being used by Naval Aviators which is obtained from nomographs in the aircraft NATOPS manual. Minor modifications are required to make this system applicable to different helicopter types. A listing of the calculator program and a discussion of the prototype's operation are included.

DD FORM 1473 (PAGE 1)
1 NOV 65

S/N 0101-807-6811



Security Classification

A-31408

| KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| LSI | | | | | | |
| Microprocessors | | | | | | |
| Helicopter | | | | | | |
| Flight Performance System | | | | | | |
| Calculators | | | | | | |

Thesis 145272
E426 Eloe
c.1
A helicopter flight
performance system using
an LSI microprocessor.